Morph Plus™

ASDG

925 Stewart Street Madison, WI 53713

MorphPlus

Cinematic Quality Visual Effects
For the Commodore Amiga

Software and Manual
by
ASDG, Incorporated

October 1, 1992
First Printing

Printed in the USA.

Contents

1 Introduction 1

1.1 What is MorphPlus? 1

1.2 The MorphPlus Package 2

1.3 Registration and Product Upgrades 2

1.4 Technical Support 3

1.5 System Requirements 4

1.6 Memory Requirements 4

1.6.1 Limiting Memory Usage 5

1.6.2 Reducing Memory Usage 6

1.7 Installation 6

1.8 Starting MorphPlus' Programs 7

1.9 About This Manual 9

2 Basic Concepts 11

2.1 Navigating the User Interfaces 11

2.1.1 User Interface Objects 11

2.1.2 The File Requester 16

2.1.3 The Chooser 17

2.1.4 The Meter Window 18

2.2 Understanding the Image Data Flow 19
2.2.1 Internal Data Types 19
2.2.2 MorphPlus' Building Blocks 21
2.3 Loading an Image 22
2.3.1 The Load Format 22
2.3.2 The Load Orientation 22
2.3.3 Image Compositing 23
2.3.4 The Image Compositing Control Panel 23
2.3.5 What a Loader Does Internally 26
2.4 Operating on an Image 27
2.4.1 The Operator Type 28
2.4.2 The Visual User Interface 29
2.5 Saving an Image 31
2.5.1 The Save Format 31
2.5.2 Type of Data to be Saved 31
2.5.3 The Save Button 31
2.6 Displaying an Image 31
2.7 Understanding Aspect Ratios 32

3 Tutorials **35**

3.1 Lesson 1: Creating a Perspective View 36
3.1.1 Setting Up 36
3.1.2 Changing the Camera Position 37
3.1.3 Picking the Point to Look At 37
3.1.4 Modifying the Image's Orientation 38
3.1.5 Showing a Preview 38
3.1.6 Performing the Perspective Scale 39

3.1.7 Viewing the Modified Image 39

3.2 Lesson 2: Rippling an Image 40

3.2.1 Setting Up 40

3.2.2 Defining a Single Ripple 40

3.2.3 Adjusting the Ripple's Behavior 41

3.2.4 Performing the Operation 42

3.3 Lesson 3: Closing a Baby's Eyes 43

3.3.1 Setting Up 43

3.3.2 Adding Vectors 43

3.3.3 Zooming In and Out 44

3.3.4 Copying and Pasting Vectors 45

3.3.5 Changing the Highlight Color 45

3.3.6 Merging Predefined Vectors 45

3.3.7 Connecting Vectors With Edges 46

3.3.8 Specifying Project Options 46

3.3.9 Starting the Warp 47

3.4 Lesson 4: Morphing An Owl Into a Baby 48

3.4.1 Setting Up 48

3.4.2 Creating a Project 48

3.4.3 Switching Between Source and Destination 49

3.4.4 Adding Vectors 50

3.4.5 Grouping Vectors 51

3.4.6 Specifying Motion and Transparency 52

3.4.7 Specifying Project Options 53

3.4.8 Creating the Morph 54

3.4.9 Morphing Between Moving Images 54

4 Using MorphPlus 57
4.1 Commands 57
4.1.1 Load Format 57
4.1.2 Load 58
4.1.3 Save Format 59
4.1.4 Save 59
4.1.5 Orientation 59
4.1.6 Compositing 59
4.1.7 About 60
4.1.8 Exit 60
4.2 Display Controls 60
4.2.1 ReDisplay 60
4.2.2 Execute 60
4.3 Color Controls 61
4.3.1 Balancing 61
4.3.2 Palette 67
4.4 Screen Controls 79
4.4.1 Horizontal Size 79
4.4.2 Vertical Size 80
4.4.3 Super Hi-Res and VGA 80
4.4.4 Dither 81
4.4.5 Colors 82
4.4.6 A-RES and A-HAM Restrictions 83
4.4.7 Other Overscan Sizes 83

5 Loaders 85
5.1 ALPHA 85

5.2 ANIM 86

5.3 IFF 87

5.4 RIPPLE 88

5.5 TEMP 90

5.6 UNIVERSAL 91

6 Savers 93

6.1 ANIM 93

6.2 HAME 95

6.3 IFF 97

6.4 TEMP 98

7 Operators 99

7.1 Apply_Map 99

7.2 Blur 100

7.3 Color_To_Gray 101

7.4 Crop_Visual 103

7.5 DCTV 104

7.6 Define_Pxl_Aspect 104

7.7 Gray_To_Color 105

7.8 Horizontal_Flip 106

7.9 KillTemp 106

7.10 Perspective 106

7.10.1 Camera Control Panel 107

7.10.2 Rotation Control Panel 108

7.10.3 Preview Control Panel 110

7.10.4 Preview Window 110

7.10.5 Output Control Panel 111
7.10.6 Other Controls 112
7.11 Refract 112
7.12 Rendered_To_Raw 114
7.13 Ripple 114
7.14 Rotate 121
7.15 Scale 123
7.15.1 Pixel Aspect 126
7.16 Sphere 126
7.16.1 Sphere Control Panel 126
7.16.2 Preview Control Panel 128
7.16.3 Preview Window 128
7.16.4 Output Control Panel 128
7.16.5 Other Controls 129
7.17 Twirl 130
7.18 Vertical_Flip 132
7.19 Warp 132
7.19.1 Vectors 133
7.19.2 Edges 138
7.19.3 Vector and Edge Restrictions 140
7.19.4 Groups 141
7.19.5 Preferences 150
7.19.6 Project Options 150

8 ARexx Interface **155**
8.1 Addressing MorphPlus 155
8.2 Getting Results of Commands 156

8.3 Launching ARexx Programs 157

8.4 Understanding the General Rules 158

8.5 Using ARexx Commands 158

8.6 Using the Supplied ARexx Programs 160

9 Morph 163

9.1 Projects 165

9.2 Onion Skin VUI 169

9.3 Frames 170

9.4 Vectors 171

9.5 Tweening 173

9.6 Preview Animation 174

9.7 Preferences 175

9.8 Project Options 176

9.8.1 Merge Control 177

9.8.2 Output Control 178

9.9 Other Options 179

10 FRED 181

10.1 Sequences, Cells, and Frames 182

10.1.1 Cells vs. Frames 183

10.2 Selection and Arrangement of Cells 184

10.3 Cell Information 185

10.4 Animated Stamps 187

10.5 The ADPro-FRED Link 188

11 AnimOps 191

11.1 Compositor 191

11.1.1 Projects 192
11.1.2 Components 192
11.1.3 Results 194

12 MorphPlus Reference 197

12.1 Starting MorphPlus 197
12.2 MorphPlus Control Screen 198
12.2.1 Commands 199
12.2.2 Balancing Controls 202
12.2.3 Palette Controls 205
12.2.4 Screen Controls 211
12.2.5 Other ARexx Commands 213
12.3 Loaders 224
12.3.1 General Information 224
12.3.2 ALPHA 226
12.3.3 ANIM 227
12.3.4 IFF 228
12.3.5 RIPPLE 229
12.3.6 TEMP 232
12.3.7 UNIVERSAL 233
12.4 Savers 233
12.4.1 General Information 233
12.4.2 ANIM 234
12.4.3 IFF 236
12.4.4 TEMP 237
12.5 Operators 237
12.5.1 General Information 237

12.5.2 Apply_Map 238
12.5.3 Blur 238
12.5.4 Color_To_Gray 239
12.5.5 Crop_Visual 239
12.5.6 DCTV 240
12.5.7 Define_Pxl_Aspect 241
12.5.8 Gray_To_Color 242
12.5.9 Horizontal_Flip 243
12.5.10 KillTemp 243
12.5.11 Perspective 244
12.5.12 Refract 248
12.5.13 Rendered_To_Raw 248
12.5.14 Ripple 249
12.5.15 Rotate 251
12.5.16 Scale 253
12.5.17 Sphere 253
12.5.18 Twirl 256
12.5.19 Vertical_Flip 257
12.5.20 Warp 258
12.6 Morph 263
12.7 FRED 268
12.7.1 Compositor 270

A Effects Samples **271**

B Troubleshooting **281**

C Hints and Tips **283**

C.1 MorphPlus 283
C.2 Warp and Morph 292

List of Figures

2.1 An example menu strip hierarchy. 12

2.2 Samples of various GUI (graphical user interface) objects. 13

2.3 The Operators chooser list. 17

2.4 Two types of meter windows. 19

2.5 The Image Information area, showing the existence of raw and rendered color image data for a 640 x 400 image. . . . 21

2.6 MorphPlus' Building Blocks. 21

2.7 The Image Compositing control panel. 24

2.8 The Image Operator area, showing the **Operator** and **Execute Op** buttons. 28

2.9 The Crop_Visual control screen, a typical user of the Visual User Interface. 30

2.10 Data flow within MorphPlus. 32

2.11 Comparison of various aspect ratios. 33

4.1 The main MorphPlus control screen. 58

4.2 A linear (neutral) color map. 62

4.3 A color map showing an increase in brightness. 63

4.4 A color map showing a decrease in contrast. 64

4.5 A color map showing an example of gamma correction. . . 65

4.6 The Balancing control panel. 66

4.7 The Palette control panel. 68

4.8 The 256 color Palette Editor. 72

4.9 The limited Palette Editor. 77

5.1 The ANIM Loader control panel. 86

5.2 The RIPPLE Loader control screen and panel. 89

5.3 Sample wave patterns with varying degrees of clipping. . . 90

6.1 The ANIM Saver control panel. The **Count Frames** button has already been selected (**Number of Frames:** is not zero), causing it to be ghosted. 94

7.1 The Blur control panel. 100

7.2 The Color_To_Gray control panel. 102

7.3 The Crop_Visual screen. 103

7.4 The Define_Pxl_Aspect control panel. 105

7.5 The Perspective Operator screen. 107

7.6 Sample image orientations. Going from left to right, top to bottom: normal, skewed, negative x, positive x, negative y, positive y, negative z, and positive z. 109

7.7 The Refract Operator screen. 113

7.8 The Ripple Operator screen. 115

7.9 Representations of unselected and selected ripple crosshairs. 116

7.10 A ripple's characteristics. 117

7.11 Adjustment of a ripple's waves to avoid unwanted results. 118

7.12 Comparison of speed versus period. Each circle represents one complete period. 119

7.13 A comparison of the three amplitude behaviors. 120

7.14 The Rotate Operator screen. 122

7.15 Components of the Rotate Circle. 122

7.16 The Scale Operator's control panel. 125

7.17 The Sphere Operator screen. 127

7.18 The Twirl Operator screen. 130

7.19 Components of the Twirl Circle. 131

7.20 The Warp Operator screen. 133

7.21 Vectors with (left image) and without (right image) edges. 138

7.22 Vector and edge placements that you should avoid creating.140

7.23 The Set Group control panel. 141

7.24 The Delete Groups control panel. 142

7.25 The Group Visibility control panel. 143

7.26 An example of how depth arrangement of groups can affect the final image. 144

7.27 The Group Depth Order control panel. 144

7.28 The Group Motion and Transparency Options control panel.146

7.29 The motion curve for the bulging eye example. 149

7.30 The Warp Preferences control panel. 150

7.31 The Project Options control panel. 151

7.32 The convex hull of a set of vectors. The right-side image was produced with Convex Hull edge type. 152

7.33 Convex Hull (left image) versus Cookie Cut (right image). 152

9.1 Components of the Morph AnimOp Screen. 164

9.2 The New Project control panels. From top to bottom: Moving Morph, Still Morph, Moving Warp, Still Warp. . . 166

9.3 The Morph vs. Warp Operations. 167

9.4 The Refresh Images control panel. 167

9.5 The Specify Frame control panel. 171

9.6 The Group Vector Tweening control panel. 174
9.7 The Group Vector Animation control panel. 175
9.8 The Morph Preferences control panel. 176
9.9 The Project Options control panel. 176

10.1 The Set Text Options control panel. 185
10.2 The Change Duration control panel. 186
10.3 The Image Info control panel. 187
10.4 The Invoke ADPro control panel. 190

11.1 The Compositor AnimOp's Components control panel. . . 192
11.2 The Compositor AnimOp's Results control panel. 195

List of Tables

2.1 Types of image data that can be merged together. 24

2.2 How mix level affects the new composite image. 25

4.1 The horizontal resolutions supported by MorphPlus. . . . 79

4.2 The vertical resolutions supported by MorphPlus. 80

4.3 Color modes and other information. The mixtures of color and display modes marked "A" are possible but require non-standard hardware or an AGA-equipped machine. MorphPlus allows these modes to be computed but will display them only if the appropriate enhanced hardware is available. 82

5.1 A list of some of the file formats detectable by the UNIVERSAL loader and the file name extensions recognized as hints. 91

7.1 Approximate indexes of refraction for various mediums. . 113

10.1 Screen depths supported by FRED. 182

12.1 Mask values for setting the screen type. 212

12.2 Dither methods and their identifiers. 213

C.1 Mix percentages to produce a time lapse effect. 291

Chapter 1

Introduction

This chapter describes the overall MorphPlus package. After reading it, you will understand:

- what is MorphPlus
- how to use the manual
- what disks and other materials are included in the package
- why it is important to register your copy of MorphPlus
- how to receive technical support
- what system hardware and software is required
- how to install the software onto your system

1.1 What is MorphPlus?

MorphPlus is an integrated set of powerful image processing tools which facilitate the creation of high quality imagery on the Commodore Amiga personal computer. MorphPlus' main qualities include:

- It can produce several video-style effects on your images, including high quality morphing, which is being used more frequently these days in movies, television ads, and music videos.
- It is fully programmable using the ARexx programming language, making it ideal for use in large projects such as the creation of animations, presentations, and large publishing works.

- It is exceedingly fast, considering the complexity of the processing being done as well as the enormous amounts of data involved. MorphPlus has been benchmarked against competing products, both commercial and freely redistributable, and was found to be 25 to 250 times faster while performing complicated processing.
- It comes from the company which pioneered color image processing on the Amiga. A company whose only focus is the development of color imaging solutions. And, a company which has been in continuous business in the Amiga market since the computer was introduced in 1985.

Although MorphPlus is a stand-alone package, current Art Department Professional (ADPro) owners can use most of the modules in this package from the same ADPro user interface that they are already accustomed to using. Please read Section 1.5 for any additional requirements for ADPro users.

1.2 The MorphPlus Package

The MorphPlus package comes complete with the following items:

- Three (3) MorphPlus disks
- This MorphPlus User Manual
- Registration Card

If you find that you are missing one or more of the above items, please contact the place where you purchased MorphPlus and ask to exchange your copy with a complete package. If they cannot offer you an exchange, please call us (our phone number is listed in Section 1.4) and we will be happy to send you a complete package.

1.3 Registration and Product Upgrades

ASDG treats the support of its products very seriously. We continually improve our products and, from time to time, make these improvements available in the form of product upgrades.

NOTE: It is absolutely imperative that you fill out and return the registration card you received with this product. If you don't, we won't be able to reach you with information about upgrades to MorphPlus.

Also, technical support will **not** be rendered to anyone who is not in our registered user database. So, please, send in your registration card.

When filling out the registration card, be sure to include the serial number of your copy of MorphPlus. MorphPlus' serial number can be found in the first **About** panel, which can be displayed by selecting the **About** button on the MorphPlus main screen. The registration card must include the serial number to be valid.

When upgrades do occur, you will have to return your original disk along with some amount of money to cover the cost of the upgrade. *International upgrades can only be processed if they are accompanied by a check in U.S. dollars drawn on a U.S. bank or by an international money order.*

Note that if you take advantage of an upgrade offer (i.e., move from one version of MorphPlus to another), the new version you receive is considered to be exactly the same software as covered by your license agreement. Therefore, you may not sell, lend, lease or give away your old version as this would be an infringement upon our copyright.

You can sell your copy of MorphPlus only if: you destroy all copies in your possession, deliver the original disks and manuals to the buyer, and inform ASDG of the transfer of ownership so that the new owner may receive technical assistance from us.

1.4 Technical Support

ASDG provides technical support via telephone at (608) 273-6585 (Monday through Friday, 10 A.M. to 5 P.M. Central Time) and upon Portal (`go asdg`), BIX (`join asdg`), and CompuServe (`go amigav`, then section 2). ASDG representatives occasionally stop in on GEnie and Usenet as well.

Additionally, you may send us your questions or comments in a letter to our business address:

ASDG, Incorporated
925 Stewart Street
Madison, WI 53713
USA

NOTE: Please remember that technical support cannot be rendered to anyone who is not in our registered user database.

1.5 System Requirements

The MorphPlus package is compatible with the entire family of Amiga computers. These include the A500, A600, A1000, A2000, A2500, A2500/30, A3000, A3000T, and A4000 Amiga computers. All programs in the MorphPlus package **require** at least Kickstart/Workbench version 2.0 and a hard drive to run. For A500, A600, A1000, and A2000 machines, an accelerator is highly recommended. If you will be using Art Department Professional with MorphPlus' modules, then you **must** have at least ADPro version 2.1.0 or higher.

1.6 Memory Requirements

By their very nature, the MorphPlus programs require a lot of memory. In fact, MorphPlus and Morph can utilize as much contiguous memory as you have on your machine. While it can run in as little as one megabyte of memory, we suggest a minimum of four megabytes of expansion memory in order to get any substantial work accomplished. Also, if you use the "high quality" features found in many parts of the program, more memory may be required.

If you use the `SETCPU` program, you should specify the `HEAD` option to allow your 32-bit memory to merge properly with your 16-bit memory. On machines with only 32-bit memory, the `HEAD` option is not required, but specifying it will not cause any problems.

Remember that fast memory must be **contiguous** for MorphPlus to make full use of it. The more contiguous fast memory you have, the larger the bit-maps (images) you will be able to process.

Ignoring the differences between the file formats that MorphPlus can process, MorphPlus really loads only two types of image: color and grayscale.

MorphPlus defines a grayscale image to be any image in which every color in the image's palette is a shade of gray (i.e., the red, green, and blue contents of each color are equal). MorphPlus considers anything which is not a grayscale image under this definition to be a color image.

To calculate how much memory would be required for a given color image, you can use the following formula as a guide:

$$MinimumBytesNeeded_{color} = Width_{pixels} * Height_{pixels} * 4$$

This guide will tell you the approximate minimum amount of *contiguous* fast memory (in bytes) required to process a color image of a given size. The multiplication by 4 to give the number of bytes required is actually a short

hand for $\frac{32}{8}$. The 32 is the number of bit-planes that MorphPlus uses for all of its processing. The division by 8 turns the number of bits into bytes (8 bits in a byte).

Take as an example an image measuring 640 by 400 pixels in size. This would require a minimum of 1,024,000 bytes of contiguous available memory to be able to take advantage of all of MorphPlus' processing capabilities. Of this space, three quarters will be used for 24 bit-plane raw image data. The remaining quarter is used to hold rendered display data.

MorphPlus converts all color images into 24 bit-plane data when the image is loaded. Grayscale image data is converted into 8 bit-planes rather than 24. This produces a memory requirement rule-of-thumb as follows:

$$MinimumBytesNeeded_{gray} = Width_{pixels} * Height_{pixels} * 2$$

The same 640 by 400 pixel image (as in the previous example) would require a minimum of 512,000 bytes of contiguous memory if it were a grayscale image rather than color.

For more information about MorphPlus' memory requirements, and for a greater understanding of how MorphPlus internally stores image data, please refer to Section 2.2.1.

1.6.1 Limiting Memory Usage

On systems with a great deal of memory, you can limit the amount of fast memory which MorphPlus will allocate. By default, MorphPlus will allocate 128,000 bytes (128K) less than the largest fast memory block available. If you wish MorphPlus to use less memory than this default, you can specify it in two ways:

1. From the Workbench, you can set a Tool Type called `MAXMEM` to the maximum number of bytes that MorphPlus will be allowed to use for its primary image buffer. For example, specifying `MAXMEM=3000000` will limit MorphPlus' primary image buffer to 3,000,000 bytes, even if you have 8 megabytes of free contiguous fast memory available.

   Tool Types can be set from the Workbench using the **Information...** command. For more information on how this is done, please consult your Amiga system software documentation.

2. From the Shell, you may specify a command line option of `MAXMEM=`*bytes*. For example, specifying the command `MorphPlus MAXMEM=3000000` will have the same effect as in the Tool Type example.

> The amount of fast memory used will be slightly larger than the amount specified as a memory limit. This extra space is used to hold the program's code in memory, as well as some small data structures.

For more information about Tool Types and Shell options, please consult Section 12.1.

1.6.2 Reducing Memory Usage

MorphPlus' enhanced palette support consumes as much as 300,000 bytes of memory. If you know that you will not make use of this feature, you can disable it and reclaim the memory it would have used.

From the Shell command line, including the `NOENHANCED` (or the shorter `NE`) option will disable enhanced palette operation.

From the Workbench, you can specify a Tool Type called `NOENHANCED=`*n* (where *n* is either zero or non-zero). If set to a non-zero value, enhanced palette operations are disabled. For example, specifying `NOENHANCED=0` permits enhanced palette operations, whereas `NOENHANCED=1` disables them.

1.7 Installation

Insert the MorphPlus Program Disk 1 into a floppy disk drive. The name of this disk is "MorphPlus_1.0_D1".

Double-click on the MorphPlus_1.0_D1 disk icon. After the disk's window opens, double-click on the `Install-MorphPlus` icon. This will start the program that will help you install MorphPlus onto your hard drive.

The Installer utility will then execute the installation script. Follow the directions given in the installation to place the MorphPlus, Morph, and FRED programs, as well as support files, where you want them.

If you plan to install MorphPlus onto floppy disks, select **Expert User** when the Installer asks you to **Set Installation Mode**. This will give you more control over where the files will be installed.

After the Installer has successfully completed the installation, remove the MorphPlus disk from the floppy drive and store it in a safe place. If your hard disk somehow becomes unreadable in the future, you will be assured of having an original working copy of the program available to you.

1.8 Starting MorphPlus' Programs

MorphPlus consists of three programs, the MorphPlus main program, the Morph program, which allows you to create a morph, and the FRED program which allows you to batch process a sequence of images. The MorphPlus program does not require Morph or FRED, but both Morph and FRED do require MorphPlus.

Starting MorphPlus

NOTE: All references to Art Department Professional (ADPro) in this manual also apply to the MorphPlus program, unless otherwise indicated. If you currently own and use ADPro, note that both MorphPlus and ADPro cannot be running at the same time—either one but not both. If you will be using ADPro, then disregard this subsection on starting MorphPlus.

You can start the MorphPlus program from either the Workbench or Shell.

For Workbench users, open the directory where the MorphPlus program is located. Locate this icon as shown to the left. It should have the name **MorphPlus** below it. Double-click on this icon to start the MorphPlus program.

For Shell users, change the current directory to the directory where MorphPlus is located and start the program with the two commands below. Type each command on its own line, hitting the `RETURN` key after each line.

```
cd MorphPlus-directory
MorphPlus
```

MorphPlus-directory must be replaced with the actual directory name where MorphPlus its files, and the Morph program were installed. Note that you can prepend the run command before `MorphPlus` above so that you can continue to use the Shell while MorphPlus is running.

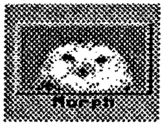
If MorphPlus is successful in initializing itself, then you will see the MorphPlus screen. If no screen is displayed, you might have run out of available memory for the program to use or the program could not find a file it needed. In either case, consult Appendix B (Troubleshooting) for more assistance.

If you start MorphPlus from the Workbench, disregard this paragraph. If you intend to start MorphPlus from the Shell, either be in the directory where MorphPlus is currently located or make sure you have previously executed the command:

`assign ADPRO:` *location_of_MorphPlus*

before starting the program. This will allow MorphPlus to find all of its support files. A good place to put this command is in your `s:user-startup` file, if it hasn't already been placed there by the installation utility.

Starting Morph



For Workbench users, open the directory where the Morph program is located. Locate this icon as shown to the left. It should have the name **Morph** below it. Double-click on this icon to start the Morph program.

For Shell users, please start the MorphPlus program before trying to start Morph. Read the first part of this section for information on how to do this. Note that you should **run** the MorphPlus program if you want to start Morph from the same Shell.

From the same shell, type the following command:

```
Morph
```

If Morph can initialize itself, then you will see its control screen (which is separate from the MorphPlus screen). If no screen is displayed, you might have run out of available memory for the program to use. Note that Morph opens a 16 color high resolution interlace screen, which does require more memory to display than the MorphPlus screen (an 8 color low resolution non-interlace screen).

Starting FRED

For Workbench users, open the directory where the FRED program is located. Locate this icon as shown to the left. It should have the name **FRED** below it. Double-click on this icon to start the program.

For Shell users, please start the MorphPlus program (if it isn't already running) before trying to start FRED. Read the first part of this section for information on how to do this. Note that you should **run** the MorphPlus program if you want to start FRED from the same Shell.

From the same shell, type the following command:

FRED

If FRED can initialize itself, then you will see its control screen (which is separate from the MorphPlus screen). If no screen is displayed, you might have run out of available memory for the program to use. Note that FRED opens a 16 color high resolution interlace screen, which does require more memory to display than the MorphPlus screen (an 8 color low resolution non-interlace screen).

1.9 About This Manual

The documentation for MorphPlus is divided into several parts. These are:

Chapter 1 Contains introductory explanations about the manual, the software, and technical support.

Chapter 2 Contains basic concepts you should be familiar with before you can try out the tutorials and use MorphPlus.

Chapter 3 Contains a few tutorials that will help you become more familiar with MorphPlus and using its controls to create the desired results.

Chapter 4 Contains the general explanation of how to use the MorphPlus program.

Chapter 5 Contains the main reference for all MorphPlus loaders.

Chapter 6 Contains the main reference for all MorphPlus savers.

Chapter 7 Contains the main reference for all MorphPlus operators.

Chapter 8 Contains general information about the ARexx Interface of the MorphPlus program. Actual ARexx commands are listed in the master reference chapter.

Chapter 9 Contains the main reference for the Morph program.

Chapter 10 Contains the main reference for the FRED program.

Chapter 11 Contains the main reference for all FRED AnimOps.

Chapter 12 Contains a master reference of all of MorphPlus' control panel commands, keyboard equivalents, menu items, and ARexx arguments.

Appendix A Contains samples done by many of the Operators in MorphPlus, as well as the Morph program.

Appendix B Contains explanations and possible solutions to several problems and questions you might encounter while using MorphPlus.

Appendix C Contains some helpful hints and tips for getting the most use out of MorphPlus.

Index Contains an index for the manual.

The manual is primarily structured as a user manual for you to read from chapter to chapter. However, it can also be used as a reference manual for the program, by using the included master reference in Chapter 12. In the early stages of learning the program, you will use the manual more as a user manual than as a reference. This will allow you to understand how the various parts of the program relate with one another. Once you become more familiar with the program, you will probably use the reference chapter more often for finding quick answers to your questions. ARexx script writers will especially find this reference chapter a necessity.

While reading through this manual, you will see four different types of text styles—plain, **bold**, `typewriter`, and *italics*. These styles are used to denote certain information.

Most of the manual is in the plain style, which denotes regular text with no particular emphasis over other material.

The **bold** style is used to denote labels (text) used in the GUI (Graphical User Interface), such as the **Execute** button on MorphPlus' main screen.

The `typewriter` style denotes something that you, as the user, would type on the Shell or in one of the input fields on the GUI.

The *italics* style is reserved, as well, for information that you need to type in. The difference is that information in this style must be defined by you. For instance, in Section 1.8 when it asks for *MorphPlus-directory*, the manual cannot put a directory name in its place because it doesn't know where you actually installed MorphPlus or the Morph program.

The *italics* style is also used to describe variables in mathematical formulas.

Chapter 2

Basic Concepts

Before you can take advantage of MorphPlus' many features, you should learn some basic concepts such as knowing how to use the program's controls, loading, saving, and displaying images, and understanding how your images are represented in memory.

This chapter will take you through the simplest steps involved in understanding these concepts. You will then build upon these concepts in later chapters.

2.1 Navigating the User Interfaces

This section gives general information about various parts of the MorphPlus' user interfaces. Basic terms such as screens, panels, buttons, and input fields will be described. The use of the file requester, chooser, and meter window will also be discussed.

2.1.1 User Interface Objects

Throughout this manual you will see references to various user interface objects, such as control screens, control panels, buttons, gadgets, input fields, cycle gadgets, rocker switches, and sliders. This subsection will give definitions of each of these, along with their use. The use of keyboard keys will also be discussed.

You should use this as a reference for all user interface terms described in this manual. If a term used in this manual has a different meaning than what will

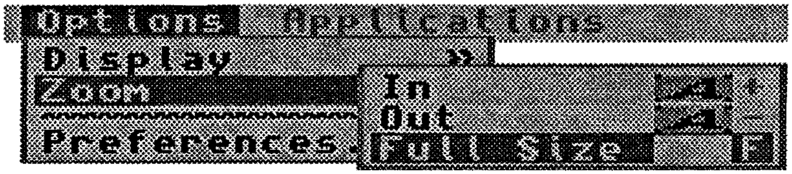


Figure 2.1: An example menu strip hierarchy.

be described below, it will be noted to you.

Consult Figure 2.2 as a reference of how some of these user interface objects look like.

Control Screens

A control screen (or screen for short) is a large area of the Amiga display. In MorphPlus, they are used to display the main controls and, for some operators, operator-specific controls. All of the screens have front-to-back gadgets, which allow you to flip between screens. Some have drag bars, which allow you to move your screen up and down to reveal other screens behind it.

Some screens have a menu bar attached to them. You can check if a screen has a menu bar by depressing the right mouse button (menu button). The menu items and subitems can be browsed by holding down the right mouse button, moving the mouse pointer up to the menu bar's (top of the screen) menus, and moving down through the menu items and subitems.

The menu bar has the following hierarchy:

menu
    menu item
        menu subitem

and a sample of this is shown in Figure 2.1.

Menus are the phrases displayed on the menu bar. Menu items are the phrases displayed below each menu. Menu subitems are the ones displayed to the right of a particular menu item (ending with ">>").

To select a menu item or subitem, move the mouse pointer above the item or subitem and release the right mouse button.

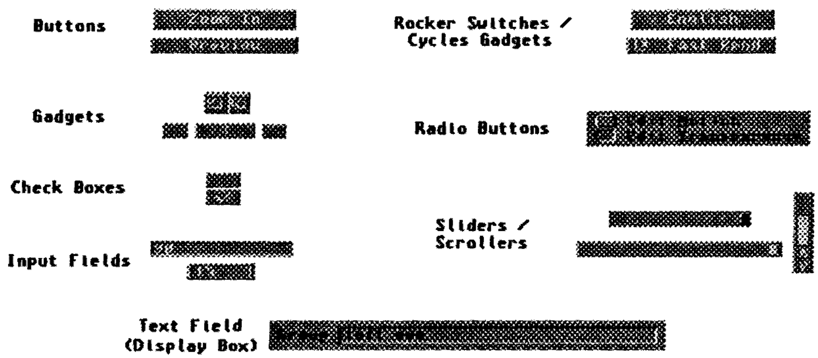


Figure 2.2: Samples of various GUI (graphical user interface) objects.

Control Panels

A control panel is a smaller area of the Amiga display that can move around on a control screen. Some panels look like standard Amiga windows (as viewable on the Workbench screen), while others have a slightly different 3-dimensional look. Some of these panels can be shrunk to a smaller size and later expanded to full size.

Buttons

Buttons are areas within windows that are outlined by a raised 3-dimensional border. Most of the time a short text string or group of strings is displayed within it. Also, whenever the left mouse button (selection button) is pressed while the mouse pointer is above this area, the button usually changes to a recessed look. In the manual, the terms "press", "select", and "click on" are used interchangeably to describe the selection of a button.

The term "double-click" is used to describe the action of selecting the button twice in rapid succession.

Gadgets

Gadgets are the same as buttons, although they describe the system's buttons—close gadget, zoom gadget, front-to-back gadget, sizing gadget, etc.

Input Fields

Input fields are areas within windows that either have a recessed 3-dimensional look or a ridged outline around them. They accept input from the user, such as a number of a text string.

Note that after you finish entering a number or string in one of these input fields, you MUST press the `RETURN` key for it to be acknowledged by the program.

For some input fields, pressing the `RETURN` key will move the cursor from field to field. Other input fields require the `TAB` key to be pressed instead. For those types of fields, pressing `Shift TAB` will move to the previous input field.

Cycle Gadgets and Rocker Switches

Cycle Gadgets and Rocker Switches are a specific, though similar, type of button. When selected, they cycle through the possible choices available for that attribute. For example, one of the buttons in the **Screen Controls** area on the main screen lets you cycle between **NonL** (non-interlace) and **ILace** (interlace).

The differences between a cycle gadget and a rocker switch are in the way they look and the way they behave. A cycle gadget uses the Workbench 2.x's imagery—an arrow pointed in a circular, clockwise direction. To cycle forward through the choices, click once on the button. To cycle backward, hold down the `Shift` key while clicking on the button.

A rocker switch looks like a regular button, but it behaves differently depending upon which horizontal half of the button you click. If you click on the right side of the button, it will cycle forward through the available choices. If you click on the left side, it will cycle backwards.

Check Boxes

A check box is a button that represents a two-state (on or off, yes or no, etc.) setting. For example, the **MRU** check box in the Chooser (see Section 2.1.3) defines whether the last chosen item from the list will be placed at the top of the list the next time the Chooser is displayed. A check mark will appear on the button for the positive choice (on, yes, etc.), whereas an unchecked box signifies the negative choice (off, no, etc.). Click once on the button to toggle between positive and negative.

Radio Buttons

Radio buttons are a set of two or more oval shaped buttons where each button represents one choice among many choices. These choices are mutually exclusive (only one can be selected at a given time). They are called radio buttons because of their functional resemblance to buttons on a radio, where each button would represent a different radio station. The currently selected choice is displayed as a recessed button. Selecting a different item will recess that item's button and un-recess ("pop up") the previous button.

Sliders and Scrollers

A slider is a gadget that lets you specify a value. A knob (rectangle within the slider area) describes the value in relation to the minimum and maximum values. For example, a slider that can specify a value between 0 and 100 will have its knob in the middle of the slider area for a value of 50. You can also use the left mouse button to select the knob within a slider and drag it to a new position. As you drag the knob, the value of the slider will be updated in real-time in the integer input or text field located to the right of the slider.

A scroller looks similar to a slider, except that the knob describes the position or amount of viewable area within a larger area. For example, the standard Amiga windows have scroller bars (including scroll arrows) to display parts of a disk or drawer's contents. To view other parts of the area, you can drag the slider knob or use the scroll arrows to view the hidden areas in smaller increments.

Sliders and scrollers can either be oriented horizontally or vertically, although most of the time (at least in the MorphPlus programs) you will see them horizontally.

Text Field / Display Box

A text field, or sometimes called a display box, is an area of a window or panel that contains a word or phrase that conveys some information to you but does not allow you to change it. Often, this field is outlined by a recessed box, signifying that the value (the text) cannot be modified. An example text field is the recessed area immediately below the list of choices in the Chooser list (see Section 2.1.3) that displays the current choice.

Keyboard Equivalents

Some of the operators have keyboard equivalents for frequently used functions. Either one key can be pressed, or a series of keys can be pressed in a particular order.

As described in Section 1.9, text written in the `typewriter` style means that it is something you can type from the keyboard. Some examples include `F10`, `Ctrl`, `Shift`, and `Del`.

If a particular equivalent requires two keys be held down, it will be listed as such: `Shift +` (with a space between the two keys. Going from left to right, you would hold down each key and press the final (rightmost) key. In the previous example, `Shift +` would mean that you should hold down the `Shift` key (either one) and press the `+` key.

Also, some control panels have "local" (only usable within the panel itself) keyboard equivalents. They are indicated by an underscore character ('_') underneath the keyboard equivalent. Instead of using the mouse to select, toggle, or activate a particular user interface object, you would simply press the keyboard equivalent.

2.1.2 The File Requester

Anytime you save or load a file, MorphPlus invokes its file requester. This subsection describes how to use the file requester to specify the name of a file to be loaded or saved.

A major part of the requester contains the list of files and subdirectories in the currently selected **Drawer**. You can select a file by clicking on its name. The selected filename will appear in the **File** input field (below this area).

Also displayed in this list area are all the subdirectories contained within the current directory. Clicking on these entries allows you to examine subdirectories. The subdirectory name will appear in the **Drawer** input field and the files within the subdirectory will appear in the list area.

If a directory contains more files and subdirectories than can be displayed in the list area, you can use the scroll bar and arrow buttons (to the right of the list area) to scroll through the current directory's contents.

To exit out of the current directory and back to its parent directory, select the **Parent** button.

Another way of selecting the directory (drawer) is to directly type it into the **Drawer** input field. Remember to press the `RETURN` key so that the file requester can acknowledge your selection.

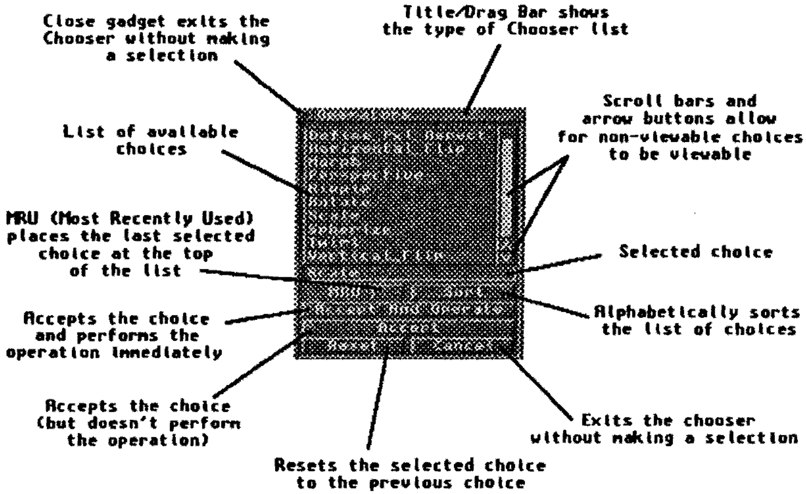


Figure 2.3: The Operators chooser list.

If you want to see all of the available disks and devices, select the **Disks** button. You can then click on one of the listed entries to examine it.

The **Close** gadget and the **Cancel** button do the same thing—they exit from the file requester without selecting a file. You would do this if you decided not to select a file at this time.

If you wanted to confirm your choice for a file, select the **OK** button. Alternatively, if the cursor is currently in the **File** input field, pressing the RETURN key will accomplish the same thing.

2.1.3 The Chooser

Several of the buttons in MorphPlus allow you to choose one item from potentially many. The current choice is displayed on the button itself. To select a different item, click once on the button to display the chooser.

The chooser opens a window (shown in Figure 2.3) and presents all of the available choices in a scrollable list box. If there are more choices than will fit in the box, you can use the scroll bar or arrows to scroll through additional items in the list. Notice that the bottommost item appears to be in an recessed box rather than an raised box like the rest of the list. This item it not actually part of the scrollable list, but rather displays which item in the list is currently selected.

You can double-click upon any item in the list. If there is a button in the chooser window marked **Accept And** *Something* (for example, **Accept And Save**), then double-clicking on an item in the list will both accept that item and start whatever action is appropriate. If the **Accept And** *Something* button does not appear in the chooser, then double-clicking on an item will simply accept that item but cause no further action.

The items in the list are kept in alphabetical order by default. However, this is not always the case. The checkbox marked **MRU** can exist in two states, either checked or not. **MRU** stands for "Most Recently Used". When checked, the item you selected and accepted the last time you saw this list will be moved from its alphabetical place and placed at the top of the list. This is especially useful when you are toggling back and forth between two or three choices in the list. Using the **MRU** feature, these two or three items will be found right next to each other at the top of the list. This makes using the chooser even faster.

Next to the **MRU** check box is a button marked **Sort**. Selecting this button causes the list to become alphabetically sorted if it had gotten out of order due to the **MRU** feature.

There will always be an **Accept** button in the chooser. When there is no **Accept And** *Something* button, the **Accept** button is the same as double-clicking on an item in the list. When there is an **Accept And** *Something* button in the chooser, the **Accept** button causes the selected item to be accepted but no further action taken.

The **Reset** button causes the item which was selected when you last entered the chooser to become active again. The **Cancel** button exits the chooser without making any choice. This is the same as selecting the close gadget at the top left of the chooser window.

2.1.4 The Meter Window

Whenever MorphPlus is processing your image, one of two styles of meter windows is displayed. Although different in appearance, both display the same information—the amount (percentage) of completeness of a particular operation.

As you can see in Figure 2.4, two general types of meter windows are used in MorphPlus. The meter at the top mimicks an old-fashioned gas gauge, where the needle displays the progress of the current operation. A description of the operation is displayed above the meter.

The second type of meter window (the bottom image) uses a "level" indicator that extends from the left to right. The description of the operation currently

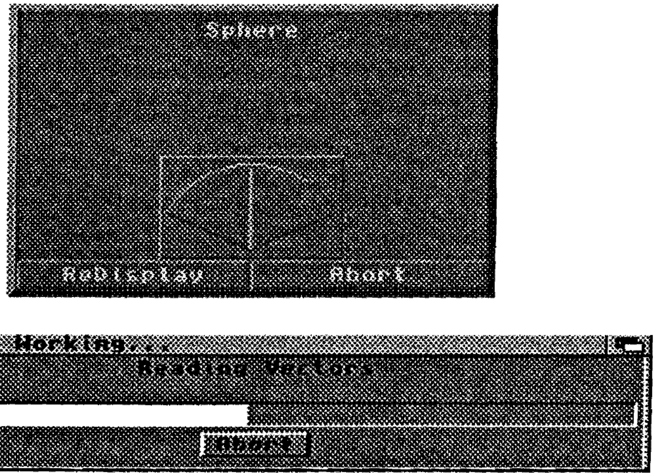


Figure 2.4: Two types of meter windows.

being performed (or of the percentage of completeness) is displayed within the meter itself.

2.2 Understanding the Image Data Flow

Understanding how image data is represented and used by MorphPlus can be very beneficial for knowing what MorphPlus is doing with your images. This section will describe how your images are stored, as well as how they "fit" within the image data flow.

2.2.1 Internal Data Types

Within MorphPlus, image data can take several forms. Understanding the differences between these data types will help you understand how and why MorphPlus works the way it does.

The first distinction is that image data can be one of two types: *raw* or *rendered.* Rendered image data is data which contains color mapped information. Images such as those which are displayable on the Amiga's screen are examples of rendered image data. For example, an IFF file which contains a 16 color image and can be displayed with any viewing program contains a color map describing

the actual color that will be displayed for each value in the image. Without the color map information, the image data can be unrecognizable.

MorphPlus can create, read, process, or write rendered data with up to 256 color mapped colors. A 256 color rendered image contains 8 bit-planes. Therefore, MorphPlus can create, read, process, or write rendered data with up to 8 bit-planes.

Raw image data is divided into color and grayscale types. Common to both types of raw image data is that a color map is not required in order to make the image data recognizable. Grayscale raw image data is stored internally in MorphPlus in 8 bit-planes (24 bit-planes, or 16.7 million shades for color).

Memory permitting, MorphPlus always converts rendered image data into raw image data during a load operation. The color map of a rendered image file is examined. If it contains only shades of gray, the rendered image data is converted into an 8 bit-plane raw gray image. If any non-gray colors are detected, the image is converted into a 24 bit-plane raw color image.

The **Execute** button causes MorphPlus to process raw image data into rendered image data.

Nearly all of the functions which MorphPlus can perform require the presence of raw image data. Only one of color or gray raw image data can be processed at one time. Operators are provided to convert between the internal formats of color and gray raw image data.

Some functions in MorphPlus require only one type of raw image data but not the other. MorphPlus will display a message if the right type of data is not currently available.

MorphPlus will display (in the lower left hand corner of the main screen under the heading **Image Information**) the types of image data available. If only rendered image data is available, the word **Rendered** will appear. If only raw color image data is available, the word **Color** will appear. If only grayscale image data is available, the word **Gray** will appear. If both raw and rendered image data is available, both Raw (**Color** or **Gray**) and **Rendered** will be indicated.

MorphPlus also tells you the width and height of the currently defined image, just below the indication of image data type. This area is called the **Image Information** area. Figure 2.5 shows an the **Image Information** area for a 640 x 400 grayscale image (both raw and rendered image data available).

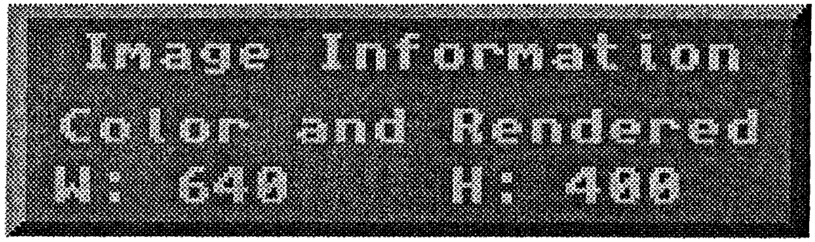


Figure 2.5: The Image Information area, showing the existence of raw and rendered color image data for a 640 x 400 image.

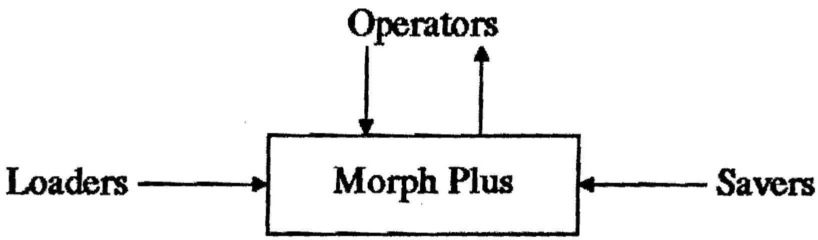


Figure 2.6: MorphPlus' Building Blocks.

2.2.2 MorphPlus' Building Blocks

The beauty of MorphPlus is that much of the program is modular, made up of separate parts that add additional functionality to the main program. These modules are called Loaders, Savers, and Operators. Figure 2.6 indicates how these modules interact within MorphPlus.

As you can see, a Loader is a source of image data.

A Saver is used as a destination for image data. Savers generally encode raw or rendered image data into a particular image file format for writing to disk.

Finally, Operators are general purpose processing modules which are used to perform the various image processing functions which MorphPlus supports. Operators are extremely flexible since they allow input, processing, and output of either raw or rendered image data.

2.3 Loading an Image

Before you can work on an image you must first load it into the MorphPlus program. MorphPlus allows you to load single IFF-ILBM images, a frame from an IFF-ANIM file, or a grayscale IFF-ILBM file to be used as an alpha channel when compositing an image over the existing image in MorphPlus' image buffer. This section will give a general description of how to do these things. If you need more information about a specific loader, please consult Chapter 5.

2.3.1 The Load Format

The Load Format button (shown on the main screen to the right of the **Load** button), displays the type of image data that will be loaded into MorphPlus the next time the **Load** button is pressed.

To change the type of image data to load, click once on the **Load Format** button. A chooser will appear (if you have enough free memory, otherwise the button will operate as a rocker switch) from which you can select a format. Once you select one, that format's name will be displayed on the **Load Format** button.

2.3.2 The Load Orientation

MorphPlus can perform a 90 degree counter-clockwise rotation of the image data *during a load operation*. This is accomplished using the **Orientation** button located on the main screen.

Note that the **Orientation** button affects data *only during a load operation.*

Using a combination of the Horizontal_Flip and Vertical_Flip operators and the **Orientation** button, MorphPlus can produce 90 degree rotations through 0, 90, 180 and 270 degrees.

These can be performed as follows:

0 Degrees — Load the image in the **Port** (portrait) orientation.

180 Degrees — Load the image in the **Port** (portrait) orientation and then perform both a horizontal and vertical flip. The resulting data will be rotated 180 degrees with respect to the data loaded from disk.

270 Degrees — Load the image in the **Land** (landscape) orientation. The resulting image will be rotated 270 degrees with respect to the data residing on disk.

90 Degrees Load the image in the **Land** (landscape) orientation and then perform both a horizontal and vertical flip. The resulting data will be rotated 90 degrees with respect to the data loaded from disk.

NOTE: Although you can use the supplied Rotate or Perspective operators to rotate to any degree (much more flexible than the previously mentioned orientation tips), these tips are included for when you want to reorient your image when it is being loaded.

2.3.3 Image Compositing

MorphPlus supports a very powerful image compositing feature which can be enabled by setting the Compositing button to **Comp**. The default value for this button is to disable image compositing. In this case, the button will be marked **Replc** (for Replace).

When the Compositing button is in the **Comp** setting and a load operation is performed, you will be presented with the image compositing control panel. Unless otherwise stated, all loaders support image compositing.

2.3.4 The Image Compositing Control Panel

If image compositing is enabled, after you select an image to be loaded, the image compositing control panel will appear, as shown in Figure 2.7.

When performing a non-compositing load operation, the loader discards the previous raw and rendered data before loading new data. When compositing, the loader merges the previously defined raw data with the new image being loaded. The exact balance of the merging operation is under your control.

A file containing a grayscale image can be merged (composited) into grayscale or color raw image data. A color image may *not* be directly merged into gray scale raw image data. Table 2.1 summarizes this restriction.

To merge a color image into a grayscale image, first convert the logical structure of the grayscale image into that of a color image using the Gray_To_Color operator.

In the Compositing control panel shown in Figure 2.7, the **Destination** sizes correspond to the width and height of the raw image currently in MorphPlus, into which the file to be loaded will be merged.

The **Source** width and height denotes the full size of the image to be loaded.

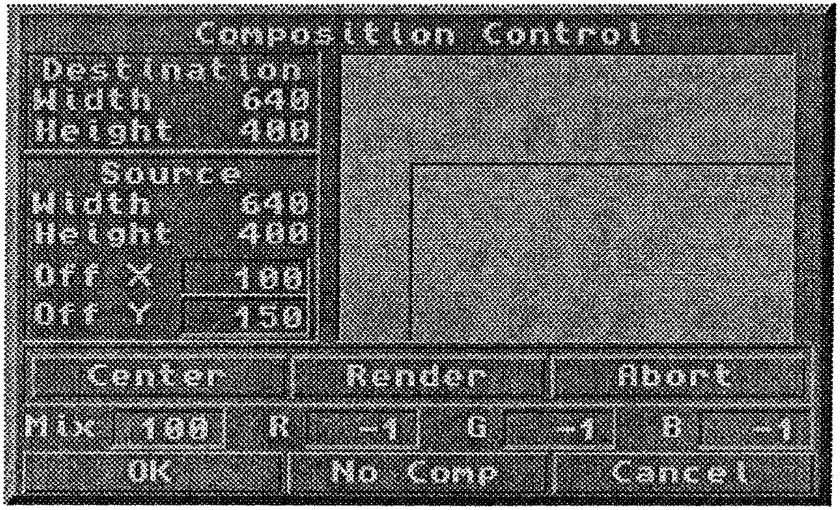


Figure 2.7: The Image Compositing control panel.

Contents Of File	Type Of Raw Data In Memory	
	Gray	Color
Gray	Yes	Yes
Color	No	Yes

Table 2.1: Types of image data that can be merged together.

Mix Level	Weight Of Pixels From Loaded Image	Weight Of Pixels From Previous Image
100	100	0
50	50	50
25	25	75

Table 2.2: How mix level affects the new composite image.

The offsets (x and y, labelled as **Off X** and **Off Y**) give the position of the top left corner of the image to be loaded relative to the top left corner of the destination image. An offset of `0,0` means that the top left corner of both the source and the destination will coincide.

Typing offsets into the supplied input fields can be used to set the offset of the top left-hand corner of the image to be merged relative to the top left-hand corner of the original image. Note that negative offsets are perfectly acceptable.

The mix level, which can range from 0 to 100, can be set by typing the desired value to be used in the supplied input field (next to the label that says **Mix**). It represents the weighting the pixels in the image to be loaded should have when being averaged with pixels from the previous (currently in MorphPlus) image. Table 2.2 shows how various mix levels affect the weighted average between these two images.

So, for example, a mix level of 100 means that the new image data is given a weighting of 100 percent. Thus, the new pixel completely replaces the old pixel where the new image and the old image overlap. A mix level of 50 means that you will get a straight arithmetic average between new and old pixels.

The labelled **R**, **G**, and **B** input fields are used to specify a 24-bit color value which will be regarded as completely transparent during the image compositing operation. One hundred percent of the old image data will be preserved for each pixel in the newly loaded data which matches the color specified in these gadgets.

When compositing a grayscale image, the value in the **R** input field is used for the transparency operation. A value of -1 in any of the input fields disables the transparency check.

Selecting the button marked **Center** causes the image now being loaded to be centered within the backdrop (previous) image. If the image now being loaded is larger than the backdrop image, its center will coincide with the center of the backdrop image.

Selecting the button marked **Render** causes the background image (the image already in memory) to be drawn in grayscale. This will assist you in determining the composition offsets. In order to display the foreground image (the one you are about to merge) in the same way, MorphPlus would have to read that image twice (once to create the mock-up for positioning, and once to actually load the image data). Loading an image from disk twice was thought to be too expensive (in terms of your time), therefore, this capability is not provided. While the background mock-up is rendering, you can stop it using the button marked **Abort**.

The button marked **OK** will merge the specified image into the currently defined raw image. The button marked **Cancel** will abort the compositing as well as the load operation.

The button marked **No Comp** will continue with the load operation but not perform compositing. This will cause the raw image to be completely replaced by the newly loaded data as if you had disabled image compositing.

Typing `Shift-RETURN` (pressing the `RETURN` key while holding down either `Shift` key) is equivalent to selecting the **OK** button. Pressing the `RETURN` key while the cursor is in either of the offset input fields will cause the cursor to alternate to the other input field. Pressing `RETURN` while the cursor is in any of the transparency or mix input fields will cause the cursor to appear in the next input field in the sequence. Typing `Alt-RETURN` (pressing the `RETURN` key while holding down either `Alt` key) causes the cursor to alternate between the offset fields and the mix and transparency fields.

With creative use of the image compositing capability, a myriad of special effects can be performed. For example, you can use a combination of a standard Amiga paint package and the image compositing function in MorphPlus to perform true color (24 bit-plane) image masking. Specifically, you can do things such as remove just one person from a scan of a group photo and place that person into the middle of a 24 bit-plane ray tracing.

2.3.5 What a Loader Does Internally

This section describes one of the most significant steps that loaders perform when reading in an image. If the file being loaded contains rendered image data (as opposed to true color or raw image data), that data will be converted into 24 bit-plane color raw image data inside MorphPlus (if the image being read is rendered grayscale, it will be converted into 8 bit raw grayscale inside MorphPlus).

Note that even a tiny 1 bit-plane image containing, for example, only red and green, will be converted into 24 bit-planes of raw data. That is, the space required to process an image can be many times larger than the space needed

by the image while it is on disk.

If there is not enough memory to convert a rendered image back into raw image data, or if you abort the conversion, no raw data will be available for you to manipulate within MorphPlus. Many of MorphPlus' advanced image processing functions require the presence of raw image data in order to operate.

However, there are some important functions you can still perform even when there is no raw image data available (either because you didn't allow it to be created or because it wouldn't fit into available memory).

If the rendered image data you just loaded is displayable by the Amiga, you can view the image directly by hitting the **ReDisplay** button. Also, you can immediately save the image out into a format which supports the same type of rendered data (i.e., IFF-ILBM or IFF-ANIM format).

From ARexx, you can force rendered image data to *not* be converted into raw image data by using the `NOPAD` loader option. This capability will be mentioned in the description of the Load Format ARexx interface.

2.4 Operating on an Image

Once the image has been loaded into MorphPlus, various manipulations can be performed on it. These manipulations come in the form of operators. The MorphPlus package comes with several operators, all of which are described in Chapter 7.

As can be seen in Figure 2.6, operators have complete freedom to read, process, and rewrite raw image data maintained within MorphPlus.

NOTE: It is important to note that unlike changes to the raw or rendered data made by any of the screen or color controls, changes made by operators to raw image data cannot be undone without reloading the raw image data.

Also, operators are not, in general, fully interruptible. Since operators actually modify the raw data maintained by MorphPlus, interrupting an operator in mid-operation will leave partially operated-upon data in memory or may, in some cases, cause all data to be lost (if interrupted).

Operators are separate programs which are run by MorphPlus when you select the **Execute Op** button located in the **Image Operator** area on the main screen. These programs must reside in a specific directory so that MorphPlus can locate them at run-time. The Installer utility that comes with MorphPlus automatically creates a directory called `Operators2` in the same directory in which MorphPlus is installed.

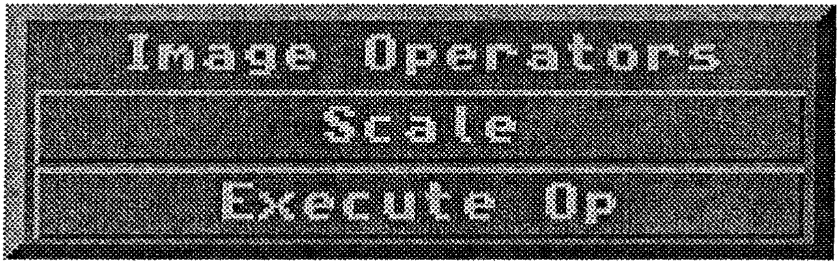


Figure 2.8: The Image Operator area, showing the **Operator** and **Execute Op** buttons.

If you execute MorphPlus from the Workbench screen, MorphPlus can automatically locate the `Operators2` (as well as the `Loaders2` and `Savers2`) directory because it is in the same directory as MorphPlus.

However, if you want to execute MorphPlus from the Shell, you must have previously executed the command:

`assign ADPRO:` *location_of_MorphPlus*

This will allow MorphPlus to properly find the `Operators2` directory.

2.4.1 The Operator Type

In the lower left corner of the MorphPlus screen, you'll see an area labelled **Image Operators** (shown in Figure 2.8). The **Operator** button (above the **Execute Op** button) is used to select which operator will be invoked when the **Execute Op** button is depressed.

Clicking on the **Operator** button will display a list of available operators. You would select the operator to use from this list. See Section 2.1.3 (The Chooser) for instructions on how to select an item from this list.

Once you have selected the operator, selecting the **Execute Op** button will cause the selected operator to execute. Alternatively, you can double-click on the operator entry in the chooser list to automatically execute it.

2.4.2 The Visual User Interface

Almost all of MorphPlus' operators use a WYSIWYG interface. WYSIWYG stands for "what you see is what you get". WYSIWYG user interfaces are characterized by a very strong visual component (hence the name "visual user interface") with immediate feedback given to the user. The different modules within ADPro which use the visual interface have common features and operating characteristics. This subsection describes these attributes.

Figure 2.9 shows the control screen for the Crop_Visual operator, which is typical of the operators that use the visual interface.

First you will note that operators which employ the VUI open their own screen in front of the MorphPlus screen. Contrast this to the non-VUI operators, such as Scale, which open a small panel on top of the main screen.

The VUI wouldn't be very visual if an image didn't play a prominent role in the screen. In Figure 2.9, you'll note that the entire left hand portion of the screen is used to display an image of the area being worked upon. This image is displayed in 8 shades of gray and is dithered to enhance the quality of the image. We chose to convert the image "on the fly" to grayscale, since we have such a small number of colors to work with on a standard Amiga display. Being usable on the lowest common Amiga configuration was a primary design goal in developing the VUI.

Note that as the image is being drawn, you can generally interrupt the rendering either by clicking on the button (or selecting the menu item, if one is available) marked **Abort** or hitting `Shift-a` (pressing the a key while holding down either `Shift` key).

MorphPlus will always scale the image so that either the height or the width will fill the screen. This allows you to see as much of the image as is possible, as large as possible. Since the Amiga's display is wider than it is tall, images with a horizontal aspect will "fill" the screen better than images with a vertical aspect. In Figure 2.9, you'll note that the image fills the height of the screen but does not fill the width. This is because the portion of the image on-screen has an overall vertical aspect.

Note that the VUI takes into account the current settings for X and Y resolution. For example, if the X resolution were set to 100 DPI and the Y resolution were set to 300 DPI, a bit-map 100 by 100 pixels in size would be drawn scaled as a stretched out rectangle. While the IFF format supplies an X and Y resolution to MorphPlus as the image is loaded, you may need to set these values yourself occasionally. See Section 7.6 for more information about setting the X and Y resolution for a given picture.

Along the top and left borders of the screen you will find a set of rulers. These rulers demark inches or centimeters, depending upon which unit of measure

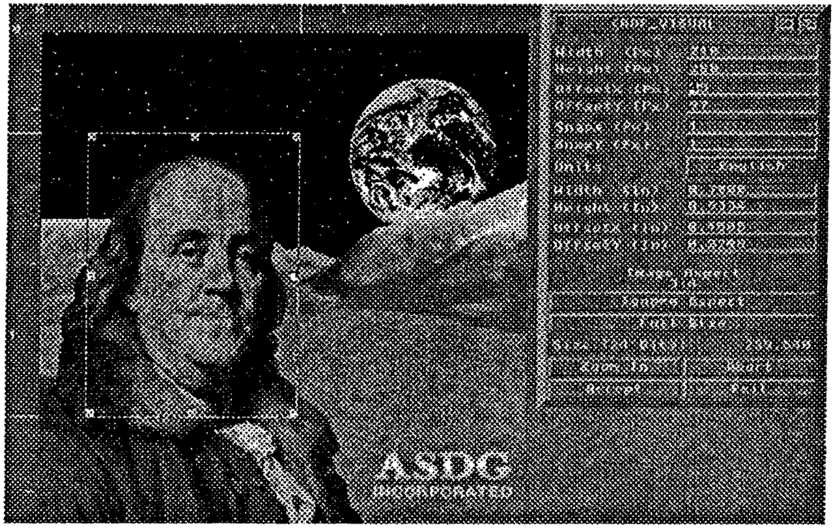


Figure 2.9: The Crop_Visual control screen, a typical user of the Visual User Interface.

you have selected. These measures, though, are dependent upon MorphPlus' knowledge of the resolution of the image. The rulers do not break down into fine detail. In inches, for example, tick marks are drawn only down to the $\frac{1}{8}$ inch level. More accurate measuring information is given in the operator's control panel.

To complete our tour of the typical VUI control screen, let's move to the top of the control panel. The button-like area at the top, in this case labeled **CROP_VISUAL**, is the drag bar for the window. By depressing the left mouse button in this area and holding it down, you can drag the entire window around the screen. You can't drag it off the screen, however. Some operators may use a different style of panel. Specifically, the panel looks like a normal window like those found on your Workbench screen. The drag bar for these windows is also at the top of the panel.

Most of the VUI operators' control panels have a "zip" gadget. You can zip the panel to a smaller size so that you can see more of the image. When the control panel is small (or zipped), you can still manipulate the objects on the image with your mouse. Some operators implement keyboard shortcuts for various commands which can be used while the control window is zipped; they are described in the specific sections in Chapter 7.

Finally, there is the front/back gadget for the screen. Use this as you would any other screen front/back gadget.

2.5 Saving an Image

When you have an image that you like to keep, you should save it to disk. You have the choice of saving it out in one of a few formats. This section will give a general description of how to do this. If you need more information about a specific saver, please consult Chapter 6.

2.5.1 The Save Format

The **Save Format** button, shown on the main screen to the right of the **Save** button, describes the types of image data that can be saved from MorphPlus and that will be save the next time the **Save** button is pressed.

As with **Load Format**, to change the type of image data to save, click once on the **Save Format** button and select the format from the chooser list. Once you select a saver, its name will be displayed on the **Save Format** button.

2.5.2 Type of Data to be Saved

Recall from Section 2.2.1 that MorphPlus can maintain up to two different types of image data internally, Rendered and Raw. Most savers will give you an opportunity to select which type of image data you wish to save. Each saver knows what type of image data it will accept. If you do not have the appropriate type of data available at the time the saver is executed, it will alert you to this requirement.

2.5.3 The Save Button

The **Save** button actually initiates the save operation. It will try to save out the current raw or rendered image data in the format you have specified in the **Save Format** button.

2.6 Displaying an Image

To display your 8 or 24-bit image data, you must create rendered image data in an Amiga displayable screen format. Due to the standard Amiga's graphics display limitation, you cannot view these 8 or 24-bit images directly on your Amiga monitor. For most current Amigas, the most colors you can display is 4096 (HAM mode). Amigas installed with the Advanced Graphics Architecture

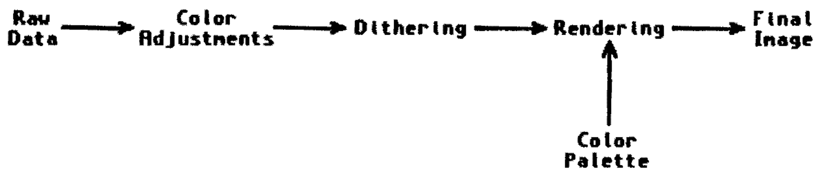


Figure 2.10: Data flow within MorphPlus.

(AGA) chip set will be able to display more colors. Consult Table 4.3 for a complete listing of available colors in various screen modes.

To create rendered image data, press the **Execute** button (located at the lower right of the main control screen). The raw image data will be analyzed and the image will start to render (display), assuming you have selected your screen controls properly. Figure 2.10 shows how MorphPlus will process the raw image data into rendered image data.

If your image is wider or taller than the viewable screen area, your image will appear to overwrite itself. This is not actually happening. It is only used to indicate that the image is larger (either wider or taller, or both) than the viewable screen area. Once the rendering has stopped, you can view hidden areas of your image by using the cursor keys. See Section 4.2.1 for more information on viewing the rendered image.

To return to the main screen, click once on the left mouse button.

2.7 Understanding Aspect Ratios

This section will explain and clarify two types of aspect ratios which you might have heard about before—image aspect and pixel aspect.

An aspect ratio is the relationship between the width of an item to its height, expressed in the form:

$$width : height$$

For example, a $1:1$ aspect ratio means that the item (whatever is being measured) is as wide as it is high.

Aspect ratios are described by saying their width value to their height value. Taking the previous example, you would say that it has a "one-to-one" aspect

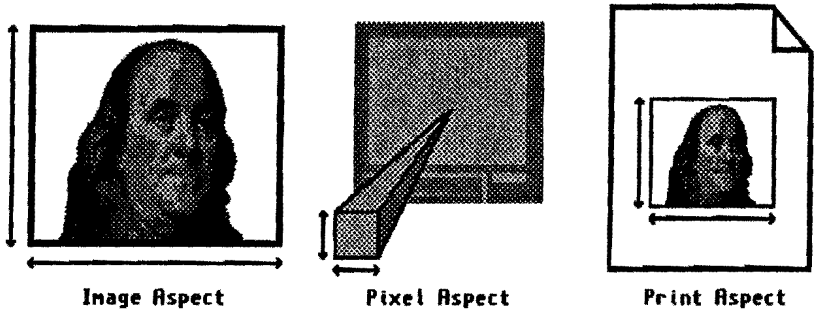


Figure 2.11: Comparison of various aspect ratios.

ratio.

Image aspect refers to the aspect ratio of the image itself. The *width* and *height* correspond to the width and height (both measured in pixels) of the image, as shown in the leftmost image in Figure 2.11. For example, a 640 x 400 image has an image aspect ratio of 640 : 400 (or more simply, 8 : 5).

Pixel aspect refers to the aspect ratio of an individual pixel on your display device (i.e., your Amiga monitor). The *width* and *height* correspond to the width and height of a single pixel, as visualized in the middle image in Figure 2.11. This ratio is different for each display device, and can also be different for similar display devices if their horizontal and vertical adjustments are different.

Chapter 3

Tutorials

This chapter contains four lessons for you to follow. Each of these lessons focuses on a different part of the MorphPlus package. After successfully completing these lessons, you should be able to use four of the operators and the Morph programs to produce wonderful results and for which you may find many uses. The following lessons are included in this manual:

Lesson 1: Creating a Perspective View

Lesson 2: Rippling an Image

Lesson 3: Closing a Baby's Eyes

Lesson 4: Morphing An Owl Into a Baby

These lessons assume that you have a basic knowledge of loading and saving images and selecting and executing operators. If you are unfamiliar with these concepts, please read Chapter 2.

3.1 Lesson 1: Creating a Perspective View

An often used visual effect is the tilting and twisting of an image as if it were floating in 3-dimensional space. This effect can be produced with the Perspective operator. Although these tilts and twists are best viewed in motion, we will create a single instance (or "snapshot") of this motion.

This lesson will take you through a typical session with the Perspective operator. After completing this lesson, you should be able to:

- Modify the size, apparent location, and orientation in 3-dimensional space of an image
- Understand how various controls affect the image
- Display a preview of an operation
- View the modified image

NOTE: Before we begin, you should be familiar with selecting loaders, savers, and operators from the main control screen, and know what various objects in the graphical user interface (GUI) do and how they to use them. If you need to refresh your knowledge, please read Chapter 2 for more information.

3.1.1 Setting Up

The first step is to load in a sample image. Select any IFF-ILBM file or pick a frame from an IFF-ANIM animation as your test image.

Once the image has been loaded, select the **Operators** button and double-click on the **Perspective** entry in the list of operators. Double-clicking automatically selects the operator and immediately executes it (so that you do not have to press the **Execute Op** button).

Perspective's control screen, titled **Perspective_Visual** will appear with a large window (**Preview**) in the upper left hand corner. This is the Preview window which shows the current size, location, and orientation based on the current Perspective settings.

As you can tell, there seems to be no way of changing your settings. This operator takes a slightly different approach from the others in that it does not automatically show all of its control panels when you execute the operator. This operator has four separate, independently-openable and closeable, control panels. They can be accessed from menu bar items.

3.1.2 Changing the Camera Position

First, open the control panel which allows you to define the camera's position by selecting the **Open...** menu item from the **Camera** menu. The camera is analagous to your view of the image. Its position in 3-dimensional space describes how far or close the image will appear and at what angle (or perspective) it will be shown.

The default **Camera Position** is (0,0,960). This means that you (the camera) are 960 units in front of the center of the image. To move closer to the image, which would make the image appear to be larger, change the **Z** value (by entering a new value or sliding the know) to a value between 700 and 800. Notice how the rectangle in the Preview window gets updated in real-time.

For this lesson, modify the **Camera Position** to these values:

X:	0
Y:	320
Z:	960

The camera is now positioned 960 units away and 320 "above" the image. The bottom part of the rectangle should now look like its slanting away from you.

3.1.3 Picking the Point to Look At

You have specified the camera position, but where is the camera pointing at? This location can also be set in the Camera control panel, under the **LookAt Point** settings. By default, the camera is always looking at the center of the image, which is located at (0,0,0). If you want the camera to "look" at a different location, change the **X**, **Y**, and **Z** values (as you did with **Camera Position**).

For this lesson, modify the **LookAt Point** to:

X:	0
Y:	107
Z:	0

From our (the camera's) position 960 units away and 320 above, we are looking at the point 107 units above the center of the image.

The other controls in this panel, **f** (focal length) and **Camera/LookAt**, will not be discussed in this lesson. Make sure that they are set to the default values of:

f:	100
Camera/LookAt:	Unlocked

3.1.4 Modifying the Image's Orientation

Now that we have defined where the camera is and where it is looking, we can now modify the image's orientation in 3-dimensional space. This is accomplished with the controls in the Rotation control panel (displayable by selecting **Open...** from the **Rotation** menu). The **Image Orientation** controls are what we are interested in.

The default values are (0,0,0) for **T**, **G**, and **P**. As a short description, **T** (short for theta) lets you rotate the image about the image's surface normal (the axis that passes perpendicular to the image plane), **G** (gamma) lets you rotate about the Y axis, and **P** (phi) lets you rotate about the axis restricted to the image plane and horizontal to the fixed X axis.

These controls can become very confusing, even for the intermediate level users. The best way to visualize how these controls will affect the rotation of your image is to try several different settings.

You can also specify which point the image will be rotated about.

For this lesson, set the Rotation values to:

T:	0
G:	0
P:	80
Rotate Pt:	**Top Center**

The top of the rectangle should now look like it's tilted away from you.

3.1.5 Showing a Preview

To see a quick preview of our modified image, we can have the Perspective operator render a preview of the image using the current settings. Open the Preview control panel by selecting the **Open...** menu item from the **Preview** menu. We will not discuss most of these controls, so make sure that they are set to:

Preview Aspect:	**No Corr**
Preview Representation:	**Rectangle**
Preview Speed:	**Fast Rend**

Select the **Preview** button to create a preview. If at any time while the image is being rendered you want to stop the rendering, press the **Abort** button.

3.1.6 Performing the Perspective Scale

Now that we have defined all of the settings for this lesson, we can actually modify the image. Open the Output window by selecting **Open...** from the **Output** menu. These controls, as well, will not be covered in this lesson. Select the **Accept** button.

The control screen will disappear and the main screen's meter window will begin showing the progress of the transformation. Once it has completed, your image will have been modified.

3.1.7 Viewing the Modified Image

To view the modified image, select the **Execute** button from the main screen. The image will begin rendering in the current **Screen Controls** settings.

To get back to the main screen, click once on the left mouse button.

This completes Lesson 1.

3.2 Lesson 2: Rippling an Image

This lesson will take you through a short session with the Ripple operator. As with many of the operators in MorphPlus, the best way to learn them is to use them.

After completing this lesson, you should be able to:

- Specify the location of a ripple, as well as how far its waves will propagate
- Control the amount of distortion that will be seen through the waves
- Know how to create parallel-looking waves across your image

---

NOTE: Before we begin, you should be familiar with selecting loaders, savers, and operators from the main control screen, and know what various objects in the graphical user interface (GUI) do and how they to use them. If you need to refresh your knowledge, please read Chapter 2 for more information.

3.2.1 Setting Up

First, select an image to use in this tutorial. Enter the Ripple operator by selecting it from the operator list. The Ripple control screen will appear, with a small grayscale representation of your image in the center. Unlike the Perspective operator, the Ripple operator always displays its control panel.

3.2.2 Defining a Single Ripple

In this lesson, we will create a single ripple across the image. Note, however, that you can just as easily create multiple ripples—with correct interference patterns among them.

The way you define a ripple is to place a ripple crosshair (shown on screen as a small crosshair with a letter next to it) at the location on the image from where the ripple's waves will propagate (radiate outward) and then define its characteristics.

The first time you use the Ripple operator a single ripple crosshair (center, or epicenter, of the ripple) is already available to be defined; it should be located in the upper left corner of your image (not of the control screen itself).

To move this crosshair, you can either enter values into the **Center X (Px)** and **Center Y (Px)** input fields or, more easily and intuitively, drag the crosshair around the screen. Position the mouse pointer above the crosshair,

hold down the left mouse button, then move the mouse around. Once you've positioned it at the desired location (which can be anything you like it to be for this lesson), release your hold of the mouse button. Did you notice how the values displayed in the control panel's input fields were updated in real-time while you were moving the crosshair around the screen? This is a feature of all operators which allow you to manipulate objects, such as these crosshairs, on a visual background (VUI).

Note that you are not limited to the area of the image itself, but anywhere outside of the image. The reason this operator allows you to specify a starting location outside of the image is to support the generation of parallel waves. By having the epicenter far enough away from the image, once the waves cross the image, they will appear to be parallel. You can try this out for your example.

3.2.3 Adjusting the Ripple's Behavior

Once the location has been set, you should define how the ripple's waves will behave. First, we'll give a short description of what this operator is actually doing. For a more complete description, please see Section 7.13.

Ripples are moving objects. This operator allows you to take a "snapshot" of these moving ripples. Each ripple has to be defined in terms of its "shape" and how it will move over time. This involves setting values for the period, speed, amplitude, and amount of decay or growth of a wave. For this lesson, we shall only concern ourselves with a couple of these attributes, namely the period and amplitude. The others can be left as an exercise that you can try out on your own.

The period of a wave is, in this operator, the length (in pixels) between corresponding points of the ripple. For example, the length from one crest (high point) to the next or from one trough (low point) to the next is the period. The longer the period, the farther apart the waves on your image, and vice versa.

The amplitude describes the height of the wave from its resting point (still water surface) to its crest, or equal to half the distance between the crest and trough. For a 2-dimensional surface (your image), the amplitude corresponds to the amount of distortion that will be evident at the crests and troughs.

Set the input fields to the following values:

Speed (Px/Frame):	10
Period (Px):	30
Amplitude (Px):	15
Phase (Deg):	0
Frame:	20
Wave Type:	**Constant**

This will create a ripple that moves 10 pixels per frame, has a period of 30 pixels, has a constant amplitude of 15 pixels over time. Frame 20 will show the ripple extending 200 (20 * 10) pixels from its center. The number of actual waves would be approximately 6 (200 / 30).

NOTE: If you are creating a ripple that starts outside of your image, change the above **Frame** *setting to a larger number. This will allow you to create a ripple large enough to stretch from the center, all the way across the image. If you don't, then the ripple will only appear outside of the image, which means that you won't be able to see it.*

3.2.4 Performing the Operation

To start creating the ripple's waves, press the **Accept** button. The control screen will disappear and the main screen meter window will start to show the progress of the operation.

Display the modified image to see what Ripple did to your image.

This step concludes Lesson 2.

3.3 Lesson 3: Closing a Baby's Eyes

Not only can the Morph program morph between two images, it can also warp a single image. This is accomplished with the Warp operator.

This lesson will take you through a session with Warp, showing how you can drop the eyelids of a baby over its eyes.

After completing this lesson, you should be able to:

- Define the areas of the image that will be warped
- Load predefined vectors
- Use the zooming feature to place vector more accurately on the image
- Use the Clipboard to copy and paste vectors
- Change the color used to represent vectors and edges
- Understand how edges affect the warped areas

NOTE: Before we begin, you should be familiar with selecting loaders, savers, and operators from the main control screen, and know what various objects in the graphical user interface (GUI) do and how they to use them. If you need to refresh your knowledge, please read Chapter 2 for more information.

3.3.1 Setting Up

The first step is to load in the picture that will be used for this lesson. If you did not install the Tutorial files at the time you installed MorphPlus, now is the time to do so. Note that you only need to install the Tutorial files (shown as an option in the installation program). The image that we will be using in this lesson is called "`babyface.pic`".

Load "`babyface.pic`", then execute the Warp operator. You should see a grayscale representation of a baby's eyes and nose.

3.3.2 Adding Vectors

To close this baby's eyelids, we must warp the skin above the eye to cover the eyeball. Let's start with the right eye.

Create a new vector by selecting the **New Vector** menu item from the **Vectors** menu. A short line with two squares at both ends of it will appear underneath your mouse pointer. By moving the mouse around, you can choose where to

place the vector. Move the vector over the right eye and press the left mouse button. This "drops" the vector down onto the image.

The vector is not in the orientation that we want it. You will notice one end of the vector is a filled square, while the other is hollow. Drag the filled square (the start point) over to the upper middle part of the eyelid (just above the eyeball). You have just specified the starting location of this particular warp area.

Now drag the hollow square (the end point) below the middle of the eyeball. The vector should be a vertical line. You can confirm this by looking at the input fields at the bottom of the screen. The currently selected vector's vector number (**Vector**), its starting (**Start X** and **Y**) and ending (**End X** and **Y**) coordinates are displayed, the change of location from start to end value (**Delta X** and **Y**), and the group of which it is a member (**Group**) are all displayed here. For a perfectly vertical line, the **Delta X** value would be 0, meaning no change in the x direction.

Also notice just below the **Vectors** input field is the display of the total number of vectors defined. Also, at the bottom right of the screen is an indication of the currently selected vector's group.

3.3.3 Zooming In and Out

The Warp operator allows you to magnify your image so that you can place vectors closer together than if you had no magnification capability. This allows you to be more exact with the placement of vectors.

To zoom into a particular area of the image, select the **In** menu subitem (from the **Zoom** menu item in the **Settings** menu). Your mouse pointer will turn into a small magnifying glass with a plus sign (+) in it. For this lesson, move this magnifying glass above the middle of the right eye and press the left mouse button. That area will be magnified, filling the same area of the screen as the previous image rendering. Note that the vector that you placed gets magnified, but the start and end points do not.

To zoom out one magnification level, select the **Out** subitem. The previous magnification will be rendered without you having to click anywhere on the image.

For this lesson, we only want to zoom in one level from normal size.

3.3.4 Copying and Pasting Vectors

To speed up the creation of similar vectors, the Warp operator supports the use of the Amiga's Clipboard for cutting, copying, and pasting vectors with the **Cut**, **Copy**, and **Paste** menu items in the **Vectors** menu.

Select the vector which you previously created. Notice how the ends of the vector become outlined. This signifies that the vector is selected. Select the **Copy** menu item or press the `Rt-Amiga C` key sequence. The vector's definition is copied to the Clipboard. Now select the **Paste** menu item (or `Rt-Amiga V`). A copy of the copied vector will now appear under the mouse pointer.

Use the Clipboard paste feature to place more vertical vectors across the right eye. You should place about 8 or 9 more vectors, with an equal amount of space between them. Also make sure that the start and end points of each vector correspond to the upper and lower parts of the eye.

3.3.5 Changing the Highlight Color

The color used to draw vectors can be changed if the vectors do not show up well on the grayscale image. Select the **Preferences...** menu item to bring up the Preferences control panel. Click on the **Primary Highlight...** button to display the Set Highlight Color control panel. Adjust the **Red**, **Green**, and **Blue** sliders until you find a color that you like. Press the **Accept** button to confirm the new color. You can also choose a secondary highlight color (**Secondary Highlight...** button) for those times when one color will not show up as well on the grayscale rendering as another color. Press the **Accept** button on the Preferences panel. You can now toggle between primary and secondary highlight color by pressing the ` (back tick) key. Pressing it once again switches the color back to the primary highlight.

3.3.6 Merging Predefined Vectors

Now that you have defined the right eye's vectors one by one, let's specify the left eye's vectors in a different way. This step will show you how you can load predefined vectors. You can either load a new set (erasing the old set) of vectors or you can merge a set with the current one. We will do the latter.

Select the **Include...** menu item (and not **Open...**) from the **Vectors** menu to display the file requester. Select the predefined vector file which contains the definitions for warping the left eye; it is named "`lefteye.vectors`" and should be located in the directory where you installed the tutorial files. If you are unclear about how to use the file requester to select a file, please read Section 2.1.2.

Notice that the vectors will appear for the left eye. They should look similar (vertical vectors) to the ones for the right eye.

3.3.7 Connecting Vectors With Edges

Vectors describe how areas of the image will be warped, but they don't, depending on their placement, describe how the points between the start points will behave. This is where edges are needed. Edges are dashed lines connecting the start and end points of two vectors. They are analagous to defining an infinite number of vectors between two vectors, but without going through the trouble of doing so.

In this lesson, we will do an experiment between the two eyes. The left eye will have edges, while the right eye won't. After we perform the warp, we should see a difference between the two.

To create an edge between two vectors, select one of the vectors, hold down the **,** (comma) key, and select the second vector. Three dashed lines will be drawn between the two vectors' start, middle, and end points, respectively.

For this lesson, select the leftmost vector on the left eye. Hold down the **,** key and successively click on the other left eye vectors, going from left to right. Do not skip any vectors in the sequence. Once all the vectors have been joined with edges, release the **,** key. The edges connecting the start points will now end up at the edges connecting the end points. If you mistakenly place an edge between two vectors, simply click on one of them, hold down the **,** key, and click on the other vector.

3.3.8 Specifying Project Options

The Warp operator lets you choose which frame of the resulting warp to actually generate. Select the **Options...** menu item under the **Project** menu to display the Project Options control panel. For our lesson, we won't discuss these controls; it will be left to the Morph lesson to discuss these options. Just set the **Total Frames** value to 10 and the **This Frame (B)** value to 10. Without getting into details, these values specify that you want the "final" effect of the warp—frame 10 of 10.

The handling of the borders of the image, as well as the edges between vectors, can be handled with the edge controls: **Outer Edge Handling** and **Edges** cycle gadgets, and the **AntiAliasing** check box. For our lesson, we will set these controls to the following:

Outer Edge Handling:	**Sliding**
Edges:	**Soft**
AntiAliasing:	checked

The above values should be adequate enough for the "every-day" warps that you will produce. To use these controls to produce different results (for the warps you create after finishing this lesson), please consult Section 7.19.6

Set the quality (**Precision**) control to the **Fast** mode. **Fast** doesn't produce the best results, but it doesn't use as much memory nor does it take as much time to complete the operation as **High Quality** mode. Of course, for the best results you should set it to **High Quality**, but for our lesson the **Fast** mode is acceptable.

Press the **Accept** button to confirm these choices and return to the Warp control screen.

3.3.9 Starting the Warp

To actually perform the warp, select the **Accept** menu item. The main screen meter window will start to show the progress of the operation. Once it has completed, you can view the warped image by pressing the **Execute** button. Notice how the baby's eyes are now closed.

Also notice how the left eyelid is smooth, whereas the right eyelid (without the edges) is ragged. If you want to be sure that the area between two source vectors moves with the vectors themselves, connect them with edges. Edges are not always needed, however. The farther two vectors are away from one another, the more the need for edges (assuming you want to move the points between the two vectors' start points in tandem).

This step concludes the warping lesson.

3.4 Lesson 4: Morphing An Owl Into a Baby

This lesson will take you through a session with the Morph program, using it to morph an owl into a baby, a transformation of which is similarly depicted on this product's box cover.

After completing this lesson, you should be able to:

- Create a morph between two still images
- Understand how to extend this lesson to deal with moving morphs (morphs between moving images)
- Add and select vectors very easily and quickly
- Categorize vectors into logical groupings
- Move between source and destination images
- Define the behavior of the vectors' movement over time
- Define the location and name of the generated frames of the morph

---

NOTE: Before we begin, you should be familiar with selecting loaders, savers, and operators from the main control screen, and know what various objects in the graphical user interface (GUI) do and how they to use them. If you need to refresh your knowledge, please read Chapter 2 for more information.

3.4.1 Setting Up

Morph is a separate program from MorphPlus, although both need to be running at the same time to generate the morph. Make sure MorphPlus is currently running, then start the Morph program.

As with the previous lesson, we will need to select the images to modify. Unlike the Warp lesson, this one will require two images—the source and destination. If you haven't done so already, install the Tutorial files with the installation utility (**`Install-MorphPlus`**). Be sure to install **only** the Tutorial files. The two files we will be using for this lesson are named "**`owl.pic`**" and "**`baby.pic`**".

3.4.2 Creating a Project

All of your work with the Morph program is through the concept of "projects". Think of a project as an individual "work in progress," which you can create,

modify, update, and throw away altogether. Although Morph allows you to create a few different project types, we will only focus on the one type that may be used quite often.

In this lesson, we will be morphing a picture of a owl into that of a baby. We will then tell the Morph program that 10 frames should be generated. This means that it will take 10 frames to transform from the source image ("`owl.pic`") to destination image ("`baby.pic`").

Select the **Still Morph...** subitem from the **New** menu item (under the **Project** menu). The Still Morph Project control panel will appear. This is where you enter the name of the project and various project-specific information. For this lesson, click on the **Select Project...** button and select the drawer (directory) which will be used to hold this project. When you create a project, Morph uses the directory you select for the project-specific files. This directory does not need to exist—the Morph program will create a directory of this name for you.

After you select the **Accept** button from the control panel, the Refresh Images control panel will appear. It is from this panel that you specify the accuracy of the on-screen representations of your source and destination images. **High Resolution** will give a better approximation of the images, although the temporary images created in the project directory will be larger than with **Low Resolution**, which sacrifices image clarity for disk space. For our lesson, make sure the **GUI Image Resolution** cycle gadget displays **High Resolution**. Also make sure that the radio button next to **Generate All** is selected. The other controls in the panel can be ignored for this lesson. After selecting **Accept** from this panel, the project directory will be initialized. Initialization involves creating a few files in the directory so that the project will be easier to use and modify. The program will display the current status of the creation process.

Note that you can create a morphing project from two sequences of images (**Moving Morph...**) or from two still images (**Still Morph...**); you can also use the **Moving Warp...** to create a warping project from a single sequence of images or **Still Warp...** from a single image. This last choice is similar to the Warp operator's function, except that **Still Warp...** lets you create a sequence instead of only one frame of a sequence (which Warp only lets you do).

3.4.3 Switching Between Source and Destination

Once the project has been created, you will be able to switch between source and destination images. The main part of the screen (the Onion Skin VUI, or Visual User Interface) displays a special onion skin type of display—both source and destination images are displayed in the same area of the screen,

but each image is given an amount of "transparency" so that you can see both images at once. This type of user interface lets you see both images at the same time which, as we will see later in this lesson, makes the process of setting up the morph more intuitive.

The horizontal slider gadget at the bottom of the screen allows you to control this transparency. If the slider knob is all the way to the left, only the source image is visible (100% of source and 0% of destination). As the knob is moved toward the right, less of the source will be visible while more of the destination will "show through". The rightmost position corresponds to 100% destination and 0% source.

Try moving this slider knob to different positions. Remember to release the left mouse button when you want to update the VUI. Notice how the display is immediately redrawn. You do not have to wait for the screen to finish redrawing before moving the knob to a new position, so feel free to experiment with different positions.

You can also move the knob using keyboard equivalents. The `c` and `v` keys let you display more of the source or destination image, respectively. Holding down the `Shift` key while pressing either key will move the knob in larger steps, while holding the `Ctrl` key will allow you to move to the leftmost (source image only) or rightmost (destination image only) position.

For our lesson, move the knob to somewhere in the middle so that you can see both images in relatively equal proportions. An easier way of doing this is by pressing the `f` key on the keyboard. We will use this onion skin "layer" in our next step.

3.4.4 Adding Vectors

A morph is created by specifying correspondences between areas of the source and destination images. These correspondences are represented on-screen as vectors and points. A vector is a line describing the movement of an area of the source image (shown as a filled square on one end of the line) toward an area of the destination (shown as a hollow end on the other end of the line). A point is an area of the image that should stay fixed (i.e., not move) during the duration of the morph. In this lesson, we will use only vectors to morph between our pictures.

First, let's outline what we want to do with the owl. As you can see on the screen, both images are visible. What we want to do is have the outline of the owl's head morph into the outline of the baby's head. By placing vectors from points on the owl's head to the baby's head, you are defining a correspondence between a feature of one image (the source image) and another (destination).

This lesson would be too involved and time-consuming if each step in the set-up procedure were listed. Instead, you will be shown how to add preexisting vectors to a work in progress. Not all of the vectors will be defined—you will be able to add vectors to parts of the image. With this approach, you will be able to learn how other people define vectors and why certain vectors were placed in particular places.

Select the **Open...** menu item (under the **Vectors** menu) and double-click on the file named "`owl2baby.vectors`". This file contains a little more than half of the vectors needed for the morph. Specifically, the head and shoulders are "tracked" (similar features between images are connected by vectors). The only thing left to do is define the vectors for the eyes.

Press `Shift` + to switch to zoom mode. With the `Shift` key depressed, the zoom will be 3:1. Click the magnifying glass mouse pointer above the left eye. At this magnification, you should be able to easily position the start and end points of the vectors between the two left eyes.

Place the first vector down on the eye; hold the `n` key while clicking on the mouse pointer above the edge of the owl's eye. This places the start point of the vector at the location of the pointer. Now, while holding down both the `n` and `,` (comma) keys, start clicking the pointer around the owl's eye, tracing it's shape. Notice how all new vectors are already connected.

Each new vector is actually a copy of the last created one. Sometimes you might find it difficult to view the contour of the eye because of too many vectors being in a small area. You might need to stop a few times (release both keys) to reposition the last created vector. After it has been changed, you can continue with this two-key process around the eye. To connect the last vector (enclosing the eye) with the first one of the group, make sure the last created vector is selected, hold down the `,` key, and then click on the first vector. Now move the end points of all of these "left eye" vectors to corresponding locations on the baby's eyes. If you find that you are unable to see the baby's eyes through the owl's, simply move the slider knob toward the right (show more of the baby picture).

Now, press the `RIGHT` cursor key a few times to move toward the right eye. Note that you do not have to wait for the screen to redraw itself before pressing the key again. This is just one of the features we put into the program so that you can spend more time working, and less time waiting.

3.4.5 Grouping Vectors

In the last section, we described vectors according to the features of the image they were modifying. The Morph program (as well as the Warp operator) allow you to assign vectors to groups. The names of these groups is up to you to

define. By default, all newly created vectors belong to the "Default" group. In this lesson, we will make three new groups, describing the three major features of the owl's face.

By holding down the **s** key and moving the mouse pointer *slowly* above all the vectors above the left eye, you can select these vectors without having to click on them. Once all of these particular vectors have been selected (you may have to pass over them a few times to "catch" them all), select the **Add Selected To Group...** menu item. The Set Groups control panel will appear, from which you can enter the name of a new group. As you can see, the initial name given to this group is "Default". This means that they were initialized as members of the "Default" group. Simply erase this name, enter the name "left eye", and press the `RETURN` key to add these vectors to the "left eye" group.

Deselect the current vectors by clicking elsewhere on the screen. Now, do the same procedure for "right eye" and "head" vectors.

3.4.6 Specifying Motion and Transparency

Morph allows you to control the motion (movement from start to end points of each vector) and transparency (amount of the source image mixed with the destination image) over time. Time can be thought of as the duration of the transformation from source to destination image, or as the number of frames that will be generated.

Select the **Options...** menu item (from the **Groups** menu) to bring up the Group Motion and Transparency Options control panel. From this panel, you can edit both the motion and transparency. Both curves are displayed on the same grid, although only one can be edited at a given time.

For this lesson, the only group that we want to modify is the eye groups. Click on the "left eye" group (in the group list at the lower left of the panel). Select the **Edit Motion** radio button to modify the motion curve for this group. To modify the shape of the curve, press the + key and click anywhere between the left and right end points. Notice how the curve immediately bends to pass through this new control point. Use the mouse to move this control point to the left of center and a little above the diagonal line (which is traced by the transparency curve).

If it is not already selected, select **Spline** from the **Interpolation Method** menu subitem. This will make a smooth curve from the start, through the middle, and terminate at the end point. The curve should be "bulging" toward the upper left side of the panel. This type of shape will cause the owl's left eye to morph into the baby's eyes at a faster rate.

To do the same change to the "right eye" curve, you can simply make a copy

of this curve. Press the **Rt-Amiga c** key sequence to copy the current group's curve definition to the Clipboard. Now, select the "right eye" group from the group list, select the **Edit Motion** radio button, and then issue the **Rt-Amiga v** (paste) sequence. Notice that the curve shape has changed to match that of the "left eye" group.

Select the **Close** menu item to return to the main Morph control screen.

3.4.7 Specifying Project Options

The generated frames of the morph will need to be saved. This, along with other options, can be specified in the Project Options control panel. Open this panel by selecting **Options...** from the **Project** menu.

For this lesson, change the **Total Frames** value to 10, **Beginning At** to 1, and **Ending At** to 10. This means that it should take 10 frames to morph from the owl to the baby, and that you want all 10 to be generated.

Change the **Finished Width** and **Height** values to the dimension of the resulting images. For most images, you would like this to match the width and height of the original images. With these images, set them to 227 and 145, respectively.

Change the other values in the control panel to the following values:

Outer Edge Handling:	**Sliding**
AntiAliasing:	checked
Precision:	**High Quality**
Preserve Source Warps:	not checked
Preserve Destination Warps:	not checked
Merge Control:	**Use Defined Transparency Map**
Saver To Be Used:	IFF
Saver Options String:	RAW
Pre Save ARexx Hook...:	none

For the **Output Sequence...** value, enter the full pathname of the resulting sequence file. Morph will create a sequence file that contains references to the generated images. This filename must have a **.seq** file extension. Enter any filename you prefer, but just remember that it must have **.seq** at the end of it.

For **Output Image Base Name...**, enter the base name of the generated images. Each generated image will have a numeric filename extension (i.e., .00001, .00002, etc.). The base name describes the part of the filename to the left of this extension. So, if you want your images to be named

`owl2baby.00001`, `owl2baby.00002`, and so on, enter `owl2babyl` (including the name of the directory or drawer prepended to it) as the base name.

Press **Accept** when you have completed all of the required definitions.

3.4.8 Creating the Morph

To create the frames that you defined in the Project Options control panel, select the **Accept** menu item. A meter window will appear atop the control screen displaying the progress of the morph. To abort the operation at any time, click on the meter window's close gadget. There will probably be a delay before your request to cancel the morph is acknowledged; Morph will finish its current task before aborting.

You can find all 10 generated frames in the directory specified by the **Output Image Base Name...** filename. To create an animation from these frames, use a suitable ANIM builder, such as the ANIM saver that comes with MorphPlus.

Alternatively, you can use the FRED program to create a stamp-size on-screen animation of this morph. Consult Chapter 10 for more information on this subject.

This step concludes the Morph lesson.

3.4.9 Morphing Between Moving Images

As a final note, you should know what would be involved in morphing between two moving images, instead of two still images as was done in this lesson. Morphing between moving images is more involved than with stills, although Morph makes it a little easier with the use of Projects.

Projects pair the corresponding frames from a moving source and destination sequence of images into a form that makes it easier to work on any frame of a morph quicker than if no project was created. Each of these image pairs are placed in a common directory, along with the vectors that will morph each frame of the source sequence to the matching frame in the destination sequence. The Morph program will load these intermediate frames faster than if you had to manually load each source and destination images for every frame that you wanted to modify—which would take too much time for complicated and multiple frame morphs.

For a moving morph (or moving warp, for that matter), the **Frame** menu's items will be enabled for you to switch among the frames of the moving sequences. To make it easier to create a set of vectors for each frame, Morph

includes the ability to tween (produce intermediate sets of vectors for a range of frames) groups of vectors. Please consult Chapter 9 for more information available for moving morphs and warps.

Chapter 4

Using MorphPlus

This chapter will describe all aspects of using MorphPlus from its graphical user interfaces, or GUIs. Detailed descriptions are included for the MorphPlus' Commands, Loaders, Savers, and Operators, as well as for the stand-alone Morph program.

As you can see in Figure 4.1, the MorphPlus control screen is made up of six areas: Commands (upper left set of buttons), Color Controls, Screen Controls, Image Operators, Image Information, and Display Controls. The first three areas and the Display Controls area will be described in this chapter; Image Operators and Image Information were discussed in Section 2.4.1.

4.1 Commands

This section describes all of the commands (shown in the **Commands** area) available from the MorphPlus main screen.

4.1.1 Load Format

The **Load Format** button displays the current image data format that will be loaded the next time the **Load** button is pressed. For a listing of available formats, please see Chapter 5.

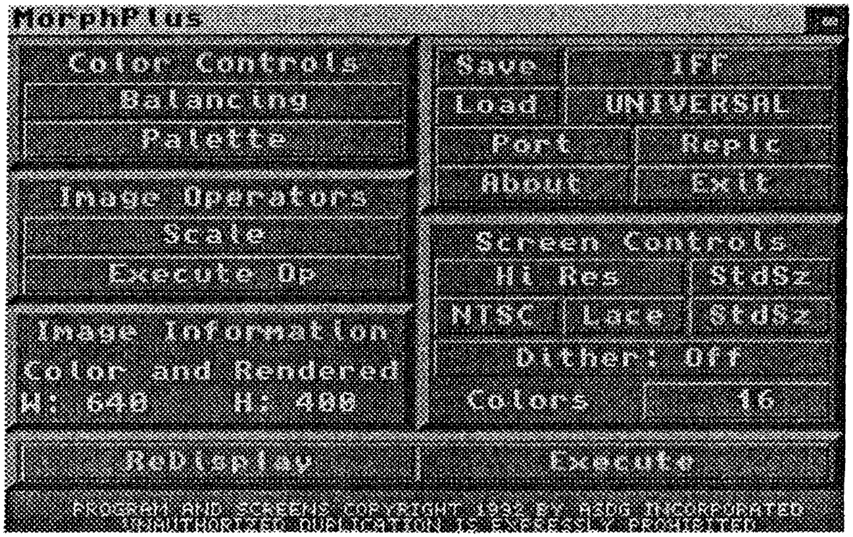


Figure 4.1: The main MorphPlus control screen.

4.1.2 Load

Aspects of the **Load** function are described more fully in Section 2.3. To summarize that information, the **Load** function in MorphPlus is actually implemented in separate programs called Loaders.

Each file to be loaded is searched for a color map. If a color map is contained in the file, it will be loaded along with the image data, provided that the MorphPlus palette is not in a **Locked** state. If the palette is **Locked**, MorphPlus will ultimately ask you if you really do wish to load the palette from the file.

If found, the palette will be inspected to see if it contains only shades of gray. If it does, the file's data will be expanded out into an 8 bit-plane grayscale image. If not, the file will be expanded out into a 24 bit-plane color image.

Depending upon the state of the **Orientation** button, the image may be rotated 270 degrees while it is being read from disk.

The expansion into 8 or 24 bit-planes is usually performed after the file has been loaded. However, if you are loading an Amiga displayable file and there isn't enough memory to hold the raw image data but there is enough memory to hold the displayable data, then it will be loaded without the conversion to 8 or 24 bit-planes. In this case, the image can be displayed, but none of the processing commands which depend on having raw data around will function.

When loading files which are already in an Amiga displayable format, the image is directly viewable immediately after loading by selecting the **ReDisplay** button.

Depending on the load format, aborting a load while in progress may or may not produce displayable image data.

4.1.3 Save Format

The **Save Format** button displays the format in which the current image data will be saved the next time the **Save** button is pressed. For a listing of available formats, please see Chapter 6.

4.1.4 Save

The **Save** button allows you to save the current raw or rendered image data in the current **Save Format**. Please refer to Section 2.5 for general information about how to save images with MorphPlus. Please refer to the Chapter 6 for information about a particular saver.

4.1.5 Orientation

The Orientation button is described more fully in Section 2.3.2. To summarize that information, the Orientation button toggles between a portrait (**Port**) and landscape (**Land**) setting. The state of this button influences the orientation of the next image to be loaded. Using this feature in conjunction with the Horizontal_Flip and Vertical_Flip operators, you can produce 90 degree rotations through 0, 90, 180, 270 degrees.

4.1.6 Compositing

The Compositing button describes whether image compositing is enabled or not. This button toggles between **Replc** (the next loaded image will replace the current image in memory) and **Comp** (the next loaded image will be composited with the current image).

Please see Sections 2.3.3 and 2.3.4 for more information about image compositing.

4.1.7 About

Selecting the **About** button will cause several information panels to appear. The information displayed includes the revision and serial number of your MorphPlus. It will also tell you the number of bytes which MorphPlus is currently using as its primary imaging buffer. Finally, it asks for your cooperation in keeping MorphPlus professionally supported.

4.1.8 Exit

Selecting the **Exit** button will cause MorphPlus to exit.

If the "don't save settings" flag is not enabled (see Sections 12.1 for a description of this), then most of the current program settings will be saved to a file called `ADProDefaults`. The next time you start MorphPlus (with the "use settings file" flag enabled), MorphPlus will try to find this file in the current directory. If it finds it, it will use the settings defined in that file to initialize the program. This allows you to retain many of the control screen and panel settings from session to session.

4.2 Display Controls

This section describes the command buttons available in the Display Controls area (located at the bottom) of MorphPlus' main screen.

4.2.1 ReDisplay

Selecting the **ReDisplay** button will cause any Amiga compatible rendered image data to be displayed. No image will be displayed if there is no rendered image data available or the available rendered image data is not in an Amiga compatible format.

If you have made any changes to the color or screen controls and have not hit the **Execute** button, the image displayed after selecting **ReDisplay** will not show the effect of any of these changes.

4.2.2 Execute

Select the **Execute** button to cause any changes you might have made in the screen, color, or image controls areas to be put into effect.

If the only change you've made is a change of screen width or height, selecting **Execute** will usually cause an image to render instantaneously.

If the image you are rendering is taller than the currently defined screen, then the image will appear to vertically wrap from bottom to top while rendering. This is normal and is MorphPlus' way of keeping you informed as to its rendering progress. Clicking on the image screen while MorphPlus is rendering will cause the MorphPlus control screen to pop to front.

When an image is being displayed, and it is larger than the currently defined screen, you can scroll about the image using the four cursor keys located on your keyboard. Scrolling can be performed in smaller increments by depressing the `Ctrl` (Control) key in conjunction with one of the cursor keys.

4.3 Color Controls

This section describes the functions located in the Color Controls area of MorphPlus. Each of the functions located in this area contributes to or affects the choice of colors which will be used in the rendering calculation.

Figure 2.10 depicts how these controls fit in to the flow of data through MorphPlus.

Notice that the color controls merely filter but do not actually modify the raw data. Therefore, you can always undo a change made to a color control and get back to the original rendered image.

4.3.1 Balancing

Whenever MorphPlus renders an image, it passes all of the raw color or grayscale data through a series of adjustments before actually using them in the rendering calculation. Before discussing the color balancing features of MorphPlus, let's first give some background information about how the adjustments are applied.

MorphPlus stores each pixel of a color image as 3 values (one each for red, green and blue) each of which can range from 0 to 255. Grayscale pixels are stored as a single value which can range from 0 to 255.

Figure 4.2 shows a linear (neutral) color map. A color map is a relationship between input intensities and output intensities. Internally, MorphPlus maintains a color map for each of the red, green, and blue components of colored data, and a grayscale map for gray data.

The color map shown in Figure 4.2 is said to be linear, or neutral, because it

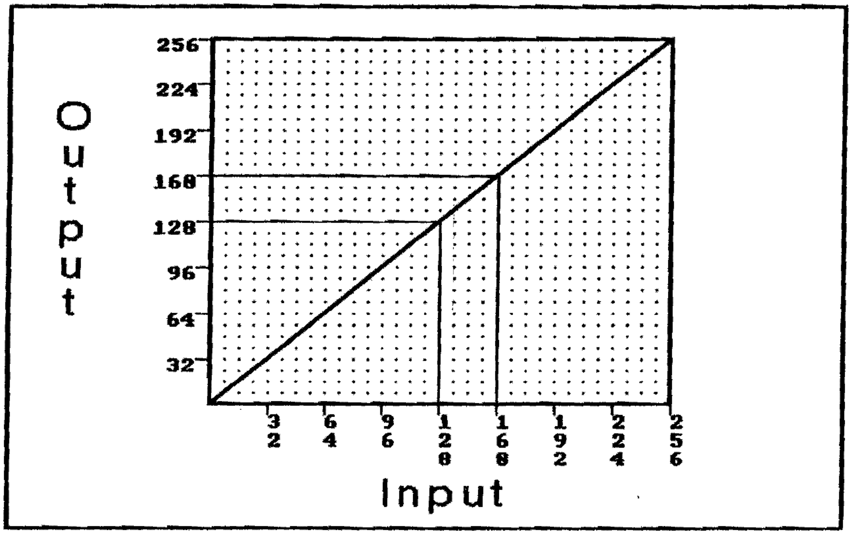


Figure 4.2: A linear (neutral) color map.

is a single straight line going from corner to corner in the color map. Notice that an incoming intensity of 128 is output unchanged as 128. The same is true for all intensities from 0 to 255; that is, they are output unchanged.

Brightness

The brightness adjustment globally modifies the general brightness of an image. It does this by uniformly shifting the color map upwards or downwards. This is shown in Figure 4.3. Here, the two input intensities are 128 and 160 (a difference of 32 intensity levels). Notice that they are output as 192 and 224 (with exactly the same difference in intensity levels). Similarly, all input intensities will be shifted upwards (or made brighter) by the color map shown in Figure 4.3.

The brightness adjustment is not without its drawbacks. Notice that an input value of 0 (in the color map shown in Figure 4.3) is output as 64. This means that the darkest intensity in the image will have an intensity of at least 64 which may not be acceptable. Also, note that all values from 192 to 255 all map to the same value—255. This means all details which had intensity levels in that range will become lost.

The brightness control in MorphPlus ranges from -50 to 50, with 0 being the neutral value. Setting the brightness control to a positive value uniformly shifts

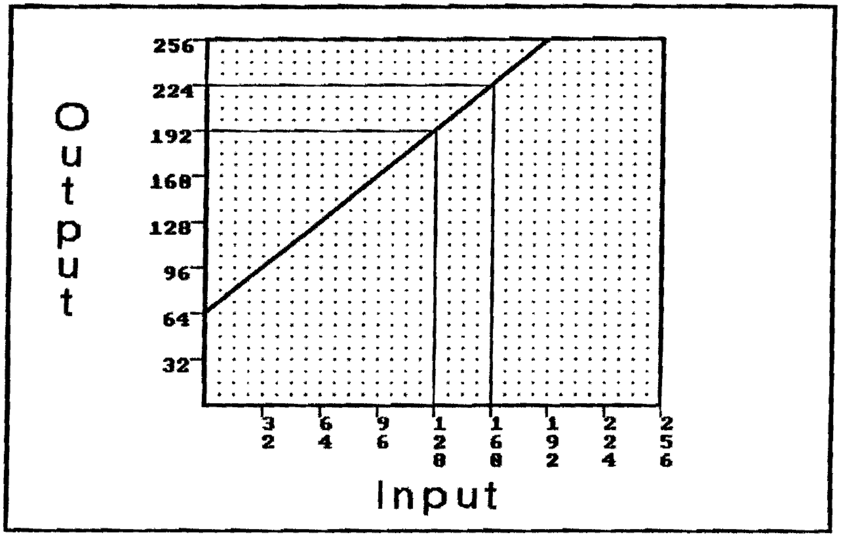


Figure 4.3: A color map showing an increase in brightness.

the color map upward (towards a brighter image). Similarly, a negative value causes the image to be shifted towards darkness.

Contrast

The contrast control globally modifies the contrast of an image. Contrast adjustments can be visualized by thinking of the neutral color map being pivoted around its center point. At one extreme, the color map becomes flat which means that all input intensities map to the same output intensity (no contrast). The other extreme is a vertical line for a color map. This produces an image with exactly two intensities (maximum contrast).

Figure 4.4 shows how input intensities 128 and 160 (a difference of 32 intensity levels) are mapped to output intensities 128 and 144 (a difference of only 16 intensity levels). Notice that the difference between two input intensities has been reduced. This produces a commensurate decrease in visible contrast.

Notice, again, that contrast loses some amount of visual detail just as the brightness adjustment. Specifically, you note that in the color map shown in Figure 4.4, input intensities which could have ranged from 0 to 255 are now restricted to the range of 64 to 192. This may or may not be acceptable for any given image.

The contrast control in MorphPlus ranges from -50 to 50, with 0 being the

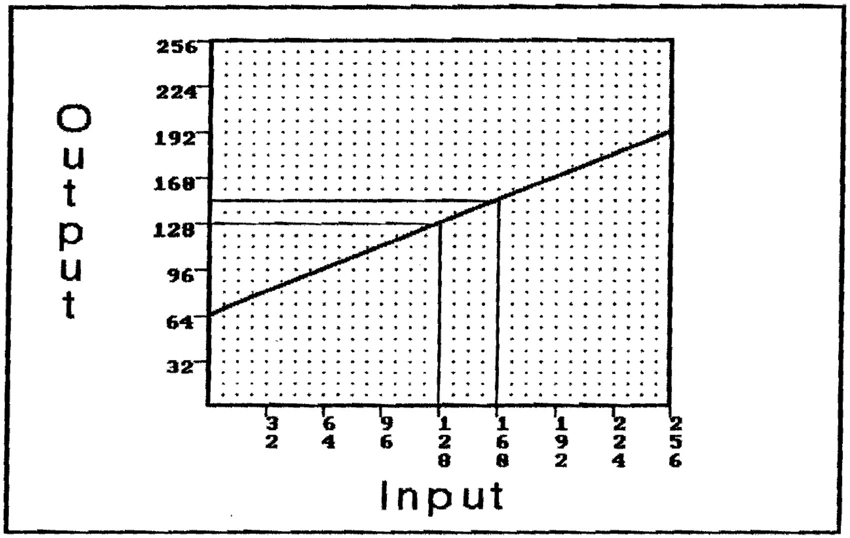


Figure 4.4: A color map showing a decrease in contrast.

neutral value. Setting the contrast control to a positive value uniformly pivots the color map around its center in counter-clockwise direction (towards the vertical) which increases visible contrast.

Gamma

The gamma adjustment provides a way to significantly brighten an image without losing much detail. It does this by introducing a curve into the color map whereby the color map is shifted upwards or downwards (made brighter or darker respectively) but no portion of the color map gets clipped to the maximum or minimum values.

Figure 4.5 shows an example color map which has a positive gamma adjustment. Notice that the two input intensities, 128 and 160 are output as intensities 192 and 216. They are, therefore, brighter.

The gamma adjustment also affects the contrast of the image. In the darker part of the spectrum, contrast is increased. However, in the lighter part of the spectrum, contrast is decreased.

The gamma control in MorphPlus ranges from -50 to 50, where 0 represents no gamma adjustment.

The overall effect of gamma adjustment is usually quite satisfactory and we

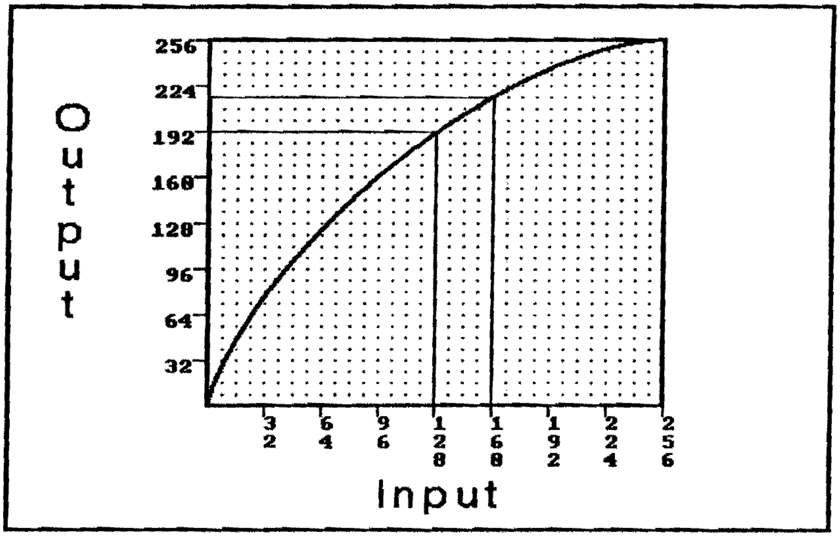


Figure 4.5: A color map showing an example of gamma correction.

recommend its liberal use.

Red, Green, and Blue Adjustments

In addition to the previously described methods of color adjustment, MorphPlus also offers control over the individual brightness of the red, green, and blue data. Grayscale brightness is directly controlled by the main brightness control so no additional control is necessary.

Having this individual control allows for simple global color balancing changes. For example, an image which has far too much red can be balanced by decreasing the global brightness of all of the red data in the image.

Note that the individual brightness controls suffer from the same drawback as the main brightness control. That is, the more heavily they are used, the more detail will be lost due to the color map being clipped against its minimum and maximum values.

The red, green, and blue adjustments in MorphPlus all range from -50 to 50, with 0 being the neutral value.

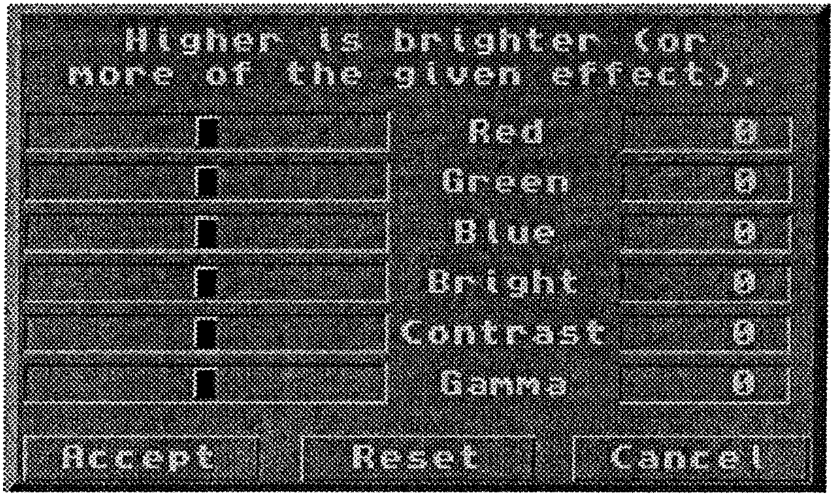


Figure 4.6: The Balancing control panel.

About the Balancing Control Panel

The panel which appears when you select the **Balancing** button (shown in Figure 4.6) contains the 6 color adjustments described above. Each is laid out as a horizontal slider.

At the bottom of the panel you'll find three buttons which will either accept, reset, or cancel any of the changes you've made to the color controls. Additionally, depressing either `Shift-RETURN` or `Alt-RETURN` will also accept any new values entered into the control panel.

Loading and Saving Balance Settings

Whenever raw image data is saved in the IFF format, MorphPlus includes the current locations of each of the balance settings. The positions of each balance setting will be restored whenever an 8 or 24 bit-plane IFF file is loaded which contains stored settings.

The saving and restoring of the positions of the balance settings are not supported in any other format other than the IFF format. However, the effect of the balance settings will be preserved whenever raw image data is saved, regardless of format.

After loading a raw image (an image which is not color mapped), you may

be presented with a panel which states that the "current balance settings do not correspond to actual data" when attempting to enter the Balancing control panel. This means that the image which was loaded did not contain MorphPlus-specific information about where to place the knobs in the Balancing control panel.

If the loaded data contained a color look-up table, that color look-up table will be loaded and effective until balance settings are "accepted". So, should you wish to use the color look-up table actually contained in a raw image, do not perform an **Accept** in the Balancing control panel. If you should enter the Balancing control panel, depress **Cancel** to preserve the color look-up table loaded from the file.

Accepting a set of balancing controls has the effect of completely overwriting any color look-up table which might have been loaded with image data.

4.3.2 Palette

MorphPlus provides very flexible and powerful palette controls. The Palette control panel is shown in Figure 4.7 and is described in this section. We'd like to mention that because the palette controls are so flexible and powerful, we cannot envision all the different effects and uses they may have. This section will describe the basic operation of the palette controls. We encourage you to experiment with the palette controls to discover new uses.

When are the Palette Controls Used?

In general, the palette controls are used in the rendering calculation only when the **Colors** button (one of the **Screen Controls**) is set to **CUST** (short for Custom).

If this button is not set to **CUST**, changes to the palette controls will not be honored, except for the palette's status and depth.

As described in the next section, a palette's status can be **Locked** or **Unlocked**. When **Locked**, the palette is "write-protected", which means that MorphPlus' own color picking technology is disabled. The effect of a **Locked** palette is in force regardless of the setting of the **Colors** button.

As described in Section 4.3.2, when MorphPlus selects colors, it can do so at two levels of accuracy. For typical applications, the **Normal** accuracy is sufficient. However, some applications, such as creating imagery for display on non-Amiga computers, may benefit from **Enhanced** accuracy mode. The palette depth selection is in force any time MorphPlus selects a color on its own.

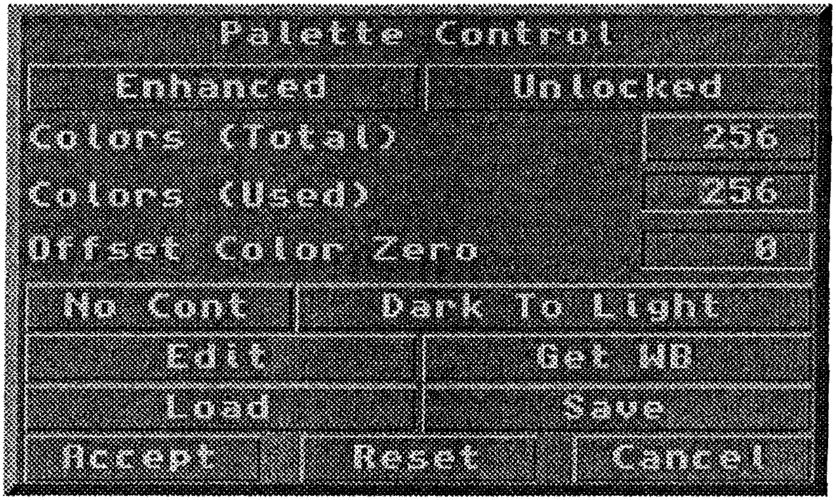


Figure 4.7: The Palette control panel.

Palette Status

The palette status button can be set to one of two values: **Locked** and **Unlocked**. This defines whether or not MorphPlus will execute its color choosing routines prior to rendering or skip directly to rendering using the pre-existing palette contents.

When the palette is **Unlocked**, MorphPlus will analyze the raw image data and catalogue various statistics about the colors it contains. It will then optimally choose a set of colors which best approximates the entire image.

When the palette is **Locked**, MorphPlus will use the prior contents of the palette and will not choose a new one. Each pixel in the image will be rendered in the color chosen from the palette which best matches the intended color.

Referring to Figure 2.10, you can see that color adjustments and dithering still affect the final image even when the palette is **Locked**.

The palette **Locked** condition can also affect image loading. If the palette is in the **Locked** state and the image you wish to load contains a palette, MorphPlus will ask you if you wish to load the palette from the file or keep the previously defined palette. This will occur whether or not the **Colors** button is set to **CUST.**

Palette Accuracy

When MorphPlus selects colors, it can do so at two levels of accuracy. Typically, the **Normal** palette accuracy is sufficient for nearly all imagery intended for use on an Amiga based computer.

Other computer systems or display technologies can take advantage of more accurate color palettes. For example, VGA systems can display up to 256 colors simultaneously chosen from an 18 bit-wide palette. Commodore's Hi-Res graphics card can display up to 256 colors chosen from a palette 24 bits wide.

Consider the difference between a palette's depth and its width (accuracy). Palette depth can be thought of as the number of colors which can be simultaneously displayed. In the case of VGA, this is 8 bits deep (256 colors). The width of the palette can be thought of as the precision with which each color can be chosen. In the case of VGA, this is 18 bits wide.

A 32 color image can be displayed on both the Amiga's own screen and on Commodore's A2410 Hi-Res graphics card. For display on the Amiga's own screen, the 32 colors are chosen from a total spectrum of 4096 choices. For display on Commodore's Hi-Res graphics card, the 32 colors can be chosen from a spectrum of 16.7 million choices.

MorphPlus' **Enhanced** palette technology carries 24 bit-plane accuracy through all color conversion computations. By offering selectable palette accuracy, MorphPlus makes it possible to prepare high quality imagery for nearly any computer display hardware including future Amiga display technologies, such as the Advanced Graphics Architecture.

The enhanced palette technology does require some additional memory. If you are in a memory limited environment, you can disable enhanced palette mode operations. If you can spare the memory, we encourage the usage of the enhanced palette mode for all picture generation.

We recommend this mode for any type of processing, except in memory limited environments because it does provide better results in almost all modes and its speed is now acceptable.

Colors (Total)

If you set the **Colors** button (on the main screen) to **CUST**, the **Colors (Total)** button takes over the function of the **Colors** button in deciding the format of the rendered image. The choices include: **2**, **4**, **8**, **16**, **32**, **64**, **128**, and **256** colors, as well as **EHB**, **HAM**, and **HAM8**.

The choice you make here determines how many colors can occupy the image's

palette. Therefore, the **Colors (Total)** button directly affects the **Colors (Used)** and **Offset Color Zero** values.

Colors (Used)

The **Colors (Used)** input field allows you to enter a number ranging from 2 to the total number of colors chosen. Normally, the number of colors used would be set to the maximum number possible for a given screen format. However, there are many situations where it is necessary to render in fewer colors than the screen format will permit.

For example, Amiga based genlocks show full color video through regions of the screen which are set to color 0. If a bitmap were to be rendered including color 0 and then genlocked, bits of genlocked video would poke through the bitmap anywhere a pixel where color 0 would be found.

To create an image which will appear solid when used with a genlock, simply don't use color 0 anywhere in the image. This can be accomplished by instructing MorphPlus to use one fewer color than is available. Then, specify a value of 1 in the **Offset Color Zero** input field. This will instruct MorphPlus to start filling in colors starting at color 1 rather than at 0. The result will be an image which contains $n-1$ (where n is the total number of colors possible) colors starting at color 1.

As another example, suppose you are required to use a specific 16 color palette and have to render an image with 4 colors (but in 4 rather than 2 bit-planes). This sort of example is a common requirement for animators. To do this, load and lock the required palette. Select a total number of colors of 16 but a number of colors to be used as 4. Specify a value in **Offset Color Zero** which will offset the colors used to the first of the 4 colors you wish to use. Note that this requires the 4 colors you wish to use to be stored contiguously in the palette.

The number in the **Colors (Used)** input field cannot exceed the total number of colors. Also, the sum of **Colors (Used)** and **Offset Color Zero** cannot exceed the total number of colors.

Offset Color Zero

The value in the **Offset Color Zero** input field is interpreted in different ways at different times. For example:

- When rendering and the palette is **Unlocked, Offset Color Zero** defines where in the palette MorphPlus will begin choosing new colors. For example, if **Offset Color Zero** is set to 4, colors 0 through 3 will not

be changed and MorphPlus will start choosing new colors beginning at color 4.

- When rendering and the palette is **Locked, Offset Color Zero** defines where in the palette MorphPlus begins fetching colors for the rendering calculation. For example, if you wanted to render a 16 color screen with only 15 colors, you can either render with the first 15 colors or the last 15 colors. Setting **Offset Color Zero** to 0 will render with the first 15 colors while setting **Offset Color Zero** to 1 will render with the last 15 colors.
- When loading a color palette, **Offset Color Zero** defines where in the palette MorphPlus will begin storing the loaded values. For example, to create a 16 color palette from two 8 color palettes, load the first 8 color palette with **Offset Color Zero** set to 0. Load the second 8 color palette with **Offset Color Zero** set to 8. Then, lock the palette, and you're ready to render.

The value in **Offset Color Zero** can be in the range of 0 to the total number of colors minus 2. Also, the sum of **Colors (Used)** and **Offset Color Zero** cannot exceed the total number of colors.

Sort Direction

When MorphPlus chooses colors and places them into the palette, it can sort them by increasing or decreasing brightness. This button is a toggle which controls the sort direction. Of particular note is that color 0 is the color which will be shown as a border around non-overscanned images. In general, you'd like this to be as dark as possible **(Darkest To Lightest)** so as not to be distracting. However, this can be overridden by selecting **Lightest To Darkest**.

Contrast or No Contrast

When MorphPlus selects colors, it orders them dark to light or light to dark. In doing so, it may place two very similar colors in color registers 0 and 1. This makes viewing and editing of the palette very difficult since most palette requesters use color registers 0 and 1 as their background and highlight pens, respectively.

If you set the Palette Contrast button to **Cont**, then MorphPlus will examine color register 0 and place a contrasting color in register 1. This eliminates the difficulty in using most palette requesters.

When the Palette Contrast button is set to **No Cont**, MorphPlus will order its palette with no special provision for distinguishing color registers 0 and 1.

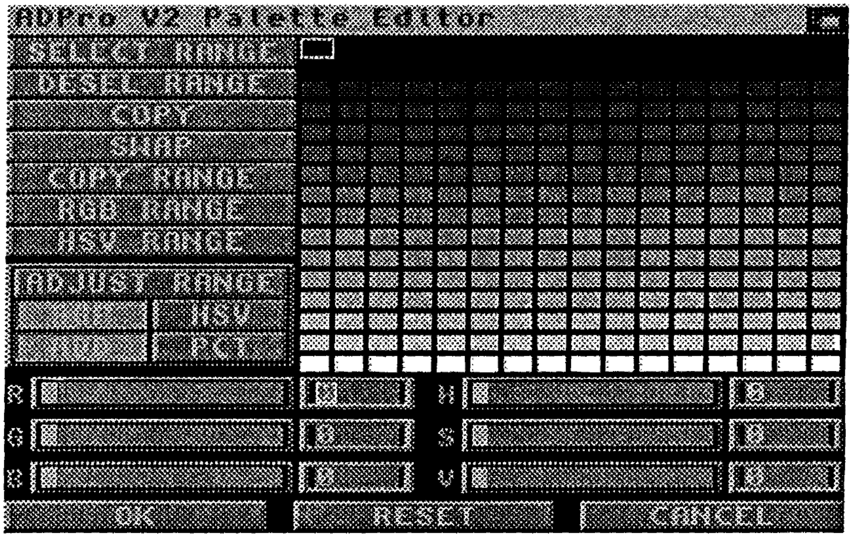


Figure 4.8: The 256 color Palette Editor.

Edit

Selecting the **Edit** button will bring up one of two color requesters. These are described below.

After editing, the effect of the new palette is immediately viewable by selecting the **ReDisplay** button in the **Commands** area. However, the image which will be displayed will simply be the previously rendered image with a new set of colors. To rerender the image (that is, go through the entire data flow as shown in Figure 2.10), you can select the **Execute** button.

The 256 Color Palette Editor

If you have enough memory, you will enjoy the benefits of being able to edit all 256 colors on-screen at one time. The new Palette Editor is accessed via the button marked **Edit** on the Palette control panel. If you do not have enough memory available to run this 256 color Palette Editor or if this editor cannot be located on disk, you will be presented with a limited palette editor which will be described later in this section.

The first thing to note is that although the Palette Editor provides access to all 256 colors on screen at one time, it does not circumvent the Amiga's 12 bit palette width. Therefore, colors which are quite close together in 24 bit space will be shown as the same color in the palette editor. For example, the color

(128, 128, 128) will be shown as the same color as (127, 128, 129). This is because most Amigas can display only 16 shades of each primary, whereas the editor allows you to edit 256 shades of each primary.

Also, hitting the `RETURN` key in the integer input fields does not advance you to the next integer input field. You can use the `TAB` to move forwards and `Shift TAB` to move backwards.

The currently selected color (the one whose values are usually shown in the **RGB** and **HSV** sliders and input fields at the bottom of the screen) is shown with a yellow highlight.

The palette editor contains many powerful capabilities for operating on ranges of colors. Colors which are part of a range are shown with either green or red highlights.

The highlights themselves can sometimes alter your perception of the colors they surround. Therefore, you can press the right mouse button at any time to cause all graphical elements on screen to fade to black—leaving only the 256 colors. (In fact, you can use the sliders at the same time the screen is blacked out by first depressing the left mouse button over the slider, then depressing the right mouse button to black out the screen).

You can select a range in two ways. First, you can select the first element of the range, then hit the button marked **SELECT RANGE**, then select the last element in the range. All of the colors between the first and last elements will be highlighted in green.

If you hold down the `Alt` key and perform the steps above, the color will be highlighted in red. The difference between red and green highlighting will be explained later.

Range deselection can be accomplished the same way. Select the first element to be deselected, hit the **DESEL RANGE** button, then select the last element to be deselected. The first and last elements, and all colors in between, will be deselected.

You can copy any color from one location to another by selecting the color to be copied, then hit the **COPY** button, then select the color to be overwritten. Two colors can be swapped in the same way (instead of hitting the **COPY** button, hit the one marked **SWAP**).

Ranges which are defined with green or red highlights are said to be **explicit** ranges (as opposed to implicit ranges).

All of the commands with the word **RANGE** in their name operate slightly differently when an explicit range is defined or not. When an explicit range is defined, the command will affect only those colors which are part of the explicit range (highlighted by red or green). When an explicit range is **not** defined,

the commands will affect all colors in between the first element selected and the second element selected.

The following example will demonstrate this:

1. Load in any color image and convert it to grayscale.
2. Select 256 colors (in the lower right corner of the screen) and hit **Execute.**
3. Enter the Palette Editor. Color register 0 will be selected yellow.
4. Hit the **SELECT RANGE** button. Notice that it stays highlighted.
5. Select color register 255 (bottom right corner). Notice all colors in between are highlighted with green.
6. Select color register 0 again (top left corner). Change this color to white using the sliders (simply move the **V** slider to 255).
7. Hit **COPY RANGE**. Notice it stays highlighted.
8. Select color register 255 (bottom right corner). Notice all colors in-between are changed to white.

Summary: *When an explicit range is defined, all members of the explicit range which are green are affected by a range command.*

1. Select a color register along the left edge of the color register grid, somewhere in the middle (of the grid's height).
2. Hit the **DESEL RANGE** button. Notice it stays highlighted.
3. Select the color register on the same line as the previously selected register, but along the right edge. Notice all of the registers in this line are no longer highlighted. They have been dropped from the explicit range.
4. Select color register 0 (top left corner). Change it to pure blue.
5. Hit **COPY RANGE**. Notice it stays highlighted.
6. Select color register 255 (bottom right corner). Notice all colors in-between have changed to blue except for those not in the range, which were unaffected.

Summary: *When an explicit range is defined, colors which are not members of the range (even if they are in the middle of the range) are not affected by range commands.*

1. Hit the **DESEL RANGE** button. Notice it stays highlighted.
2. Select color register 0. All green highlighting should now be gone. You should have a color grid with all pure blue except for one line of pure white.

3. Select the first white color.
4. Hit **SELECT RANGE.**
5. Select the last white color. Notice all of the white colors are now highlighted with green.
6. Select color register 0 (top left corner). Make this color pure red.
7. Hit **COPY RANGE.**
8. Select color register 255 (bottom left). Notice that only the colors in the explicit range were affected.

Summary: *When an explicit range is defined and a portion of the range is contained between the end points of a range command, only those colors in the range are affected by range commands.*

1. Select the first highlighted color.
2. Hit the **DESEL RANGE** button.
3. Select the last highlighted color. No color should be highlighted with green at this point.
4. Select color register 0 (upper left corner) which should be pure red.
5. Hit **COPY RANGE.**
6. Select the color register still containing blue immediately before the first register containing red. Notice all color registers in between turn red.

Summary: *When no explicit range is defined, all range commands work on an implicit range defined by the selected color before the command button is hit...and the color selected after the command button is hit.*

If you select a color register or range of color registers with either `Alt` key held down, the register or range of registers will be highlighted with red instead of green.

Colors highlighted in red are part of the **explicit** range but their color values are locked. They will contribute a "place holder" to range commands, but their colors will not be affected by range commands.

This is best explained with the **RGB RANGE** and **HSV RANGE** commands and another example:

1. Ensure that no color registers are currently highlighted with red or green (i.e., no explicit range defined).
2. Select color register 0. If it isn't already, make it pure red.
3. Select color register 15 (top line, right corner). Make it pure blue.
4. Hit **RGB RANGE** (notice it stays selected).

5. Select color register 0. Notice that color registers 0 to 15 now contain a ramp from red to blue done in 16 steps.
6. Set colors 4 through 11 to pure white.
7. Define an explicit range of colors 0 to 15 by selecting 0, hitting the **SELECT RANGE** button, and selecting color 15.
8. Select color 4. Hit the **SELECT RANGE** button. Hold down the **Alt** key and select color 11. You should now have an explicit range defined from 0 to 15 with colors 4 through 11 being highlighted in red and the others highlighted in green.
9. Select color register 0. Hit **RGB RANGE**. Select color 15. *You should see nothing change.* This is because you computed a red to blue ramp with 16 steps, same as before...and it didn't affect the colors highlighted in red.
10. Select color register 4 (the first white color). Hit the **DESEL RANGE** button. Select color register 11 (the last white color). You should be left with 8 green highlighted colors separated into two groups of four, with 8 highlighted colors inbetween.
11. Select color register 0. Hit **RGB RANGE**. Select color register 15. Notice that the 8 white colors are unchanged but the 8 highlighted colors will change. They still contain a ramp from red to blue but the ramp is now done in 8 steps, not 16.

Summary: *Having colors in an explicit range highlighted with red means they contribute a place holder to range commands, but are not affected by them.*

The difference between **RGB RANGE** and **HSV RANGE** is that the **RGB RANGE** command builds a ramp through the RGB color space (while the **HSV RANGE** command uses the HSV space).

The **HSV RANGE** command is very handy for computing ranges in which all colors have the same intensity but have different hues, etc.

The **ADJUST RANGE** command allows you to adjust the color balance of an entire range of colors simultaneously. To use the **ADJUST RANGE**, you must have an explicit range defined. Select which type of adjustment you wish to make, either in RGB space or HSV space and either ADDitive or PCT (percentage), then hit the **ADJUST RANGE** button. After you are done adjusting, hit the **ADJUST RANGE** button again to exit the adjusting mode.

The Limited Palette Editor

As mentioned earlier, if you don't have enough available memory to display this 256 color Palette Editor or if it cannot be found on disk, you will be give a limited palette editor to use.

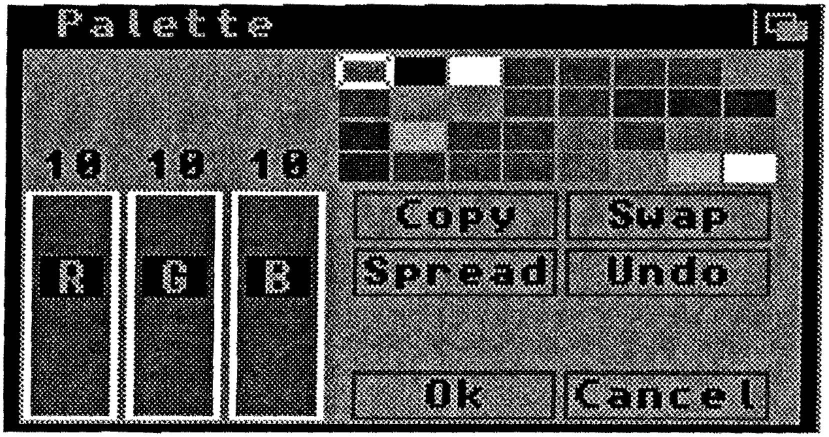


Figure 4.9: The limited Palette Editor.

You may manipulate one color at a time, chosen by clicking the left mouse button over one of the colored boxes at the right hand top of the color requester. These boxes represent the colors contained in each color register. The number of boxes displayed will correspond to the number of user definable colors in the screen mode you have selected with the **Colors** button.

The color which you have chosen to manipulate will be rendered in a rectangular area in the top left hand portion of the color requester.

Along the left hand edge of the color requester, three vertical sliders can be found which correspond to the red, green, and blue composition of the color being modified.

Below the color boxes (at the right middle of the color requester) are four command buttons. These are defined below.

Selecting the **Copy** command will duplicate the current color into the next color register you click with the left mouse button. This will cause the two color registers to have the same value. The current color register will become the register you copied to.

Selecting the **Swap** command will exchange the contents of the current color register with the contents of the next color register you select with the left mouse button. The current color register will become the register you swapped to.

Selecting the **Undo** command will restore the previous state of the palette, undoing any previous **Swap**, **Copy**, or **Spread** command. The current color register is unchanged.

Selecting the **Spread** command will uniformly place colors between the current color register and the next selected register. The color registers you select become end points. Any color registers in between them are changed to a color in between the two end points.

Load

Selecting the **Load** button causes the file requester to appear. Using it, you can select a file from which MorphPlus will attempt to read a palette. All Amiga format images, such as 2, 4, 8, and 16 color images, have color palettes. However, raw image data, such as 18, 21 or 24 bit-plane files, do not. Palettes can also be loaded from IFF "brush" files as well.

Palette loading is affected by several other factors in the Palette control panel. Specifically, if the number of colors in the palette to be loaded exceeds the number of colors available to be loaded into, the excess colors will be ignored.

For example, if the total number of colors permissible at the time **Load** is selected (as defined by the value in the **Colors (Used)** input field) is 15, and the palette to be loaded contains 32 colors, only the first 15 will be loaded.

In fact, if the value in **Offset Color Zero** is non-zero at the time **Load** is selected, then the number of colors which will be loaded (in the example cited in the previous paragraph) will be lower than 16.

Note that a loaded palette will be overwritten if the palette is not in a **Locked** condition and you select the **Execute** button to render an image.

Save

The currently defined palette can be saved to a file by selecting the **Save** button. Upon doing so, the file requester will appear. After selecting a filename, the palette will be stored to disk.

Note that the saved file will contain only a palette. It will not contain any image data. Therefore, use caution when selecting a pre-existing file to store the palette, as that file will be overwritten with palette information. Its previous contents will be lost.

Get WB

This command is very useful for creating imagery which will be displayed on the Amiga Workbench. Selecting this command will cause MorphPlus to load the currently defined Workbench colors into the palette starting at the color

Mode	Width
Low Res	320
Low Res/Overscan	368
Hi Res	640
Hi Res/Overscan	736

Table 4.1: The horizontal resolutions supported by MorphPlus.

defined by **Offset Color Zero**.

This command fetches *up to four colors* found in the Preferences structure. Kickstart 2.x allows the Workbench to contain more than four colors. However, these colors are not found in the Preferences structure as defined by earlier versions of the operating system.

Therefore, consider the **Get WB** button as a short hand for loading only the first four colors of the Workbench. Should you require access to the palette of a more than four colored Workbench (possible under Kickstart/Workbench 2.0 or later), use the palette loading facility to load a palette from the `ENV:sys/palette.ilbm` palette preferences file or other pre-saved palette file.

4.4 Screen Controls

MorphPlus' screen controls offer a selection of 208 possible video modes, as many or more than any other Amiga program at this time. Once an image has been fully rendered, many of the changes from one screen mode to another are instantaneous.

NOTE: Changes to any of the screen controls do not affect the raw or rendered data in any way. Therefore, such changes can be quickly undone.

4.4.1 Horizontal Size

The various horizontal sizes are shown in Table 4.1. Note that some color modes preclude some horizontal sizes. These limitations are shown in Table 4.3.

Given a fully rendered image, changes from low resolution to high resolution will be instantaneous if the color mode chosen is allowed in both low and high resolution.

Mode	Height
NTSC	200
NTSC/Overscan	240
NTSC/Laced	400
NTSC/Laced/Overscan	480
PAL	256
PAL/Overscan	296
PAL/Laced	512
PAL/Laced/Overscan	592

Table 4.2: The vertical resolutions supported by MorphPlus.

4.4.2 Vertical Size

The various vertical sizes are shown in Table 4.2. All color modes are supported in all vertical sizes. Selecting an NTSC vertical size on a PAL machine simply truncates the screen size at the NTSC lower boundary. Selecting a PAL vertical size on an NTSC machine simply extends the screen size below the visible lower border of your NTSC screen.

Given a fully rendered image, changes from one vertical size to another are instantaneous. Aspect correction for changing between interlaced and non-interlaced screens can be accomplished using the digital scaling capability of MorphPlus.

4.4.3 Super Hi-Res and VGA

If you are running on an Amiga equipped with the Enhanced Chip Set, then several new screen modes become available. These are:

- Super Hi-Res (1280), which allows double the horizontal resolution normally allowed by Hi-Res, including the Hi-Res overscan mode (1472).
- VGA which allows 480 or 960 lines vertically (depending upon the state of the interlace flag).

Be aware, however, that these modes have some significant restrictions (including the fact that they may contain only 4 colors chosen from a palette of only 64 choices). These modes are supported for completeness and future compatibility.

4.4.4 Dither

Dithering is a technique for achieving greater color fidelity at the expense of spatial fidelity (image sharpness). MorphPlus supports six dithering types. These are: None, Floyd-Steinberg, Burkes, Sierra, Jarvis, Stucki, and Random dithers.

With the exception of the Random dither, the dithers are presented in the order of greatest to least effect on the image. However, counter to intuition, they range from fastest to slowest.

Floyd (1) (Floyd-Steinberg) is the tightest of the dithers and produces good results for all video modes. The **Stucki (5)** dither is the sparsest (and most time consuming to compute) and produces good results where just a little dithering is desired.

Dithers 1 through 5 are each "error diffusion" dithers. The added computation time of the higher numbered dithers results from incorporating a greater number of pixels into each dithering computation. This also accounts for why the higher numbered dithers affect the image less. That is, the error diffusion is spread over a larger number of pixels (thus affecting each pixel less).

The Random dither is useful in the preparation of successive images which will become part of an animation. While the other dithering techniques produce superior results, they may also produce an undesired flickering when multiple images are animated. The Random dither does not have this problem when animating.

After selecting the dithering style of your choice, and any other modifications to the image using MorphPlus' other image processing capabilities, select the **Execute** button (located in the lower right hand corner of the screen) to render the image.

The **Dither** button will activate the Dither Type chooser. See Table 12.2 for a listing of available dithers.

NOTE: Dithering is applied during rendering and has no effect on the raw 8 bit-plane grayscale data or 24 bit-plane color data. Therefore, you may change and rechange the dithering setting at will without affecting the original data.

Dithering gives its best results when displayed in the high resolution video modes. Dithering provides a benefit in the HAM modes but can slightly increase the amount of HAM fringing present in the rendered image. Of the dithering choices available, the Random dither is the least likely to introduce HAM fringing but is the weakest dither.

Setting	Colors	Bitplanes	Low Res	Hi Res
2	2	1	Y	Y
4	4	2	Y	Y
8	8	3	Y	Y
16	16	4	Y	Y
32	32	5	Y	A
64	64	6	A	A
128	128	7	A	A
256	256	8	A	A
EHB	64	6	Y	A
HAM	4096	6	Y	A
HAM8	262144	8	A	A

Table 4.3: Color modes and other information. The mixtures of color and display modes marked "A" are possible but require non-standard hardware or an AGA-equipped machine. MorphPlus allows these modes to be computed but will display them only if the appropriate enhanced hardware is available.

4.4.5 Colors

The various choices of color mode are given in Table 4.3. This table also indicates the restrictions that the choice of color mode places on the horizontal size.

The color mode is selected using the **Colors** button. Clicking on this button will display the Number of Colors chooser list. See Section 2.1.3 for more information about the general use of chooser lists.

Notice that some entries in Table 4.3 indicate that that particular mode will not work in either low or high resolution Amiga screen modes. This indicates a number of colors which is not supported by the display capabilities of most Amigas. When rendering an image in 64 to 256 colors, no image will be displayable on the Amiga's screen (unless you are running the program on an AGA-equipped Amiga). Rendered image data is still available, however, for display on non-standard display devices such as the Commodore A2410 Hi-Res graphics card or for saving using a Saver which supports that particular number of colors.

For a detailed description of the **CUST** color mode setting, please see Sections 4.3.2 and 4.3.2.

4.4.6 A-RES and A-HAM Restrictions

The A-HAM data format is a variation on Sliced HAM (SHAM) popularized by Rhett Anderson, formerly of Compute Magazine. In A-HAM, a new set of 16 base color registers is chosen for each line in the image. This increases the color fidelity of a HAM image at the expense of increased machine and display overhead.

MorphPlus will read A-HAM images in the same file format (as registered with CATS, Commodore Applications and Technical Support) used by NewTek's Dynamic HAM image files.

The A-RES modes are 16 color hi-res variants developed by ASDG, Incorporated. The A-RES file format was designed to be backwards compatible with NewTek's Dynamic Hi-Res format as registered with CATS.

A-RES allows all 4096 colors supported by the Amiga to be present on one hi-res screen at the expense of extreme processor overhead while an A-RES image is being displayed.

NOTE: The A-RES modes are not for general purpose use as are the standard Amiga display modes. There are many limitations to what can be done while displaying an A-RES image and not all images lend themselves to rendering in this mode.

The generation of A-RES and A-HAM images is not fully supported. While MorphPlus will continue to be able to read these images, it can no longer generate them. This decision was reached after careful consideration based upon knowledge of future developments.

4.4.7 Other Overscan Sizes

Tables 4.1 and 4.2 provide the dimensions of the overscan screens directly supported by MorphPlus. These values were suggested by the hardware documentation provided by Commodore. However, the Amiga can produce horizontal and vertical overscans larger than the values directly supported by MorphPlus.

If you wish to produce an image conforming to a different overscan size, simply render the image to the desired size (since MorphPlus will render at any size) and use a program other than MorphPlus for display purposes.

Chapter 5

Loaders

This chapter provides detailed descriptions of the use and operation of all loaders which are included with MorphPlus. ARexx control of these loaders is described in Section 12.3.

5.1 ALPHA

The ALPHA loader is an IFF-ILBM loader that allows you to use a grayscale IFF-ILBM image as an alpha channel, controlling the transparency of the image to be loaded and composited over the current image in MorphPlus' image buffer.

Currently, the file specified as the alpha channel must be a grayscale IFF and must have the same width and height as the image being loaded. In this alpha channel file, the whiter the pixel the more the foreground (loaded image's) pixel will replace the background (current image's) pixel, and the darker the pixel the less it will replace it. You can create alpha channel files in any paint program for use as sophisticated masks or artistic effects.

First, you need to have your "background" image in MorphPlus. If you need to load an image, be sure to use a non-ALPHA loader. Next, make sure you have the compositing button on MorphPlus' main control screen set to **Comp**. Select the ALPHA loader, then press the **Load** button. A file requester will appear for you to select the image to composite. Once you have selected one, the Image Compositing control panel will appear. If you don't want to load an alpha channel file, then use the **No Comp** button, otherwise press the **OK** button. Another file requester will appear, this time asking you to select the

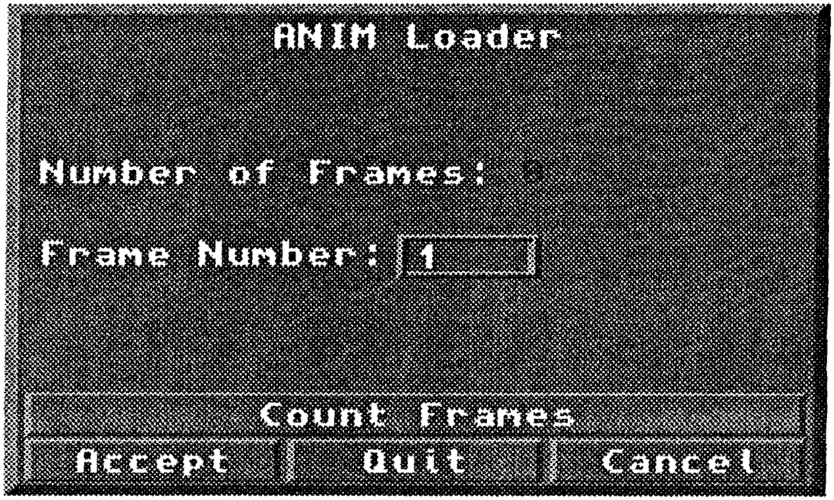


Figure 5.1: The ANIM Loader control panel.

alpha channel file. Be sure that the file you select is a grayscale image with the same width and height as the composited image, or else the mix levels defined in the Compositing control panel will be used (instead of the alpha channel file). If all requirements are met, the image will be composited using the alpha channel you specified.

Be sure you do not delete the `.ALPHA` and `ALPHA` files from the `Loaders2` directory. The ALPHA loader needs both of these files to operate. These files are placed in this directory by the installation program.

5.2 ANIM

The ANIM loader allows you to load any frame within a properly defined IFF-ANIM (Op Mode 5) animation file.

After selecting the **ANIM** loader and pressing the **Load** button, a file requester will appear for you to select an ANIM file. Once a valid IFF-ANIM file has be selected, the ANIM loader control panel will appear, as shown in Figure 5.1. The loader will then ask you for the specific frame number (shown in the **Frame Number:** input field) to load. Assuming that the animation has that frame number, it will load that frame into MorphPlus.

The next time you load an ANIM frame, the number specified in the **Frame**

Number: input field will be incremented, assuming you will want to load the next frame in the animation.

Just above the **Frame Number:** input field is the **Number of Frames:** indicator. It will display **0** until you press the **Count Frames** button, which will count how many frames are in the ANIM file and display this value.

A time-saving feature of the ANIM loader is that it remembers the position of the last frame you selected to load. This allows you to load successive frames without having to wait for the loader to search for that frame. For animations made up of a few frames, this feature might not be as evident as with a "multi-frame" animation.

To exit the loader without actually loading a frame, select **Cancel**. The control panel will disappear, returning control back to the main screen. If you want to use the time-saving feature just described, select this button to exit the control panel.

If you want to exit the control panel without saving the information about the last frame loaded, select the **Quit** button. **Quit** will close the ANIM file, allowing you to use it in other programs.

NOTE: If you want to play the animation with an ANIM viewer, you MUST select the **Quit** *button after finishing all of your work (or exit the MorphPlus program). If you fail to do this step, the ANIM viewer will not be able to read the file.*

5.3 IFF

The IFF loader (also referred to as the Super-IFF loader) allows you to load any properly defined IFF-ILBM picture file, including all of the following:

- Commodore standard IFF files in 1 through 5 bit-planes, corresponding to 2 through 32 colors, respectively.
- Commodore standard IFF files in the 64 color Amiga Extra-HalfBright (EHB) format.
- Commodore standard IFF files in the 4096 color Hold-And-Modify (HAM) format.
- IFF files in Sliced-HAM (SHAM) format (can be read but not written).
- IFF files in A-HAM or Dynamic HAM formats (4096 colors).
- IFF files in A-RES or Dynamic Hi-Res formats (4096 colors in hi-res).

- Commodore standard IFF files in 12, 15, 18, 21, and 24 bit-planes (4096, 32768, 262144, 2097152, and 16777216 colors, respectively). This includes 21 bit-plane files created by Digi-View 4.0.

The Super-IFF loader will look for a palette in the image file. If there is one, it will be tested to see if it contains only grayscales. If it does, the image will be loaded into 8 bit-planes and will be treated as grayscale image data. Otherwise, the image will be loaded into 24 bit-planes and treated as a color image.

When attempting to load an image already in a rendered format (such as the 1 to 6 bit-plane formats) and there isn't enough memory to convert the full image into 24 bit-planes or 8 bit-planes if grayscale), the Super-IFF loader will simply load the image and not convert it into a deep bit-plane format. This means you will be able to view the image, edit its palette, and even save the image back out in a file format supporting the same depth, but you will not be able to perform any other of MorphPlus' image processing functions.

The IFF loader fully supports image compositing. For more information about compositing, please refer to Section 2.3.3.

5.4 RIPPLE

The RIPPLE loader allows you to generate grayscale IFF-ILBM alpha channel files from rippled patterns. It is similar to the Ripple operator, except that it does not modify an existing image, but that it creates a new grayscale representation of the ripple patterns themselves.

NOTE: The effect of ripples is best explained in the Ripple operator section of the manual (Section 7.13). Please read that part of the manual before using this loader. Only those parts that are different between the two will be discussed here.

After selecting the **RIPPLE** loader and pressing the **Load** button, the control screen and panel (as shown in Figure 5.2) will appear.

The first difference you will see between the operator and loader is that the loader's control screen does not contain an image. The Ripple epicenters you will be defining will actually create the image.

The width and height (in pixels) of the resulting image is defined in the **Image X (Px)** and **Image Y (Px)** input fields, respectively. The center square on the control screen background represents the dimensions of the image, although changes to these fields will not change the proportions of this square, but will change those of the wavefront indicators (assuming **Wavefront** is set).

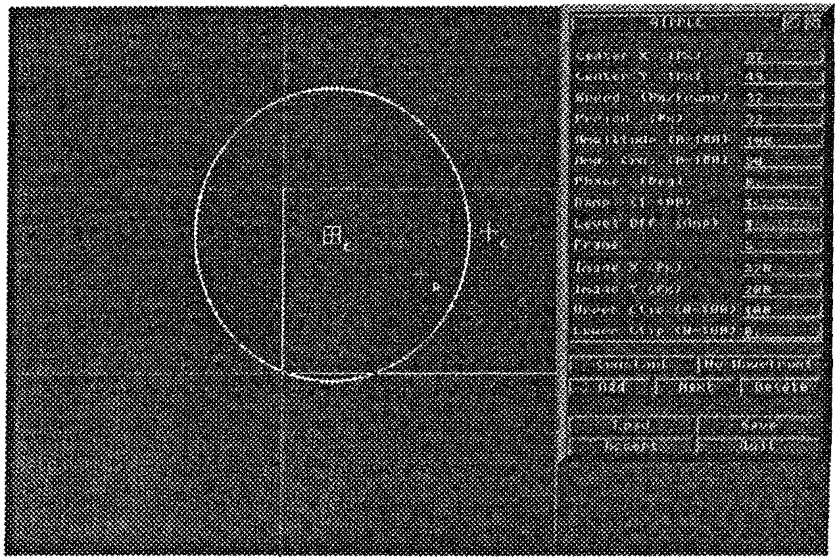


Figure 5.2: The RIPPLE Loader control screen and panel.

Almost all of the ripple characteristics are the same as in the operator. The main difference is with the amplitude settings. In the Ripple operator the amplitude of a wave is described in pixels, whereas in the loader it is a percentage (0 to 100). Grayscale intensities can range between total black (0) and total white (100). The ripple patterns that will be generated are described according to this range. You can think of these intensities as wave altitudes. The whiter parts of the ripple describe the waves' crests, whereas the darker parts represent the troughs.

The "still water" level of the wave (**Amp. Cen. (0-100)** input field) is specified as a number between 0 and 100. This allows you to define how light or dark the resulting wave patterns will be. It is from this level that the amplitude (also measured as a number between 0 and 100 in the **Amplitude (0-100)** input field) will be measured.

For example, an amplitude center of 50 and amplitude of 50 will produce a wave pattern that cycles through the range of 0 to 100. If you were to change the center to 75 but keep the same amplitude, the range would become 25 to 100. Note that the wave's height cannot exceed 100 nor be less than 0. This clipping may produce long bands of all white or all black, depending upon which end of the range is closer to the center.

There might be circumstances when you don't want to produce these all black or all white bands but do want bands to occur (i.e., you want banding to

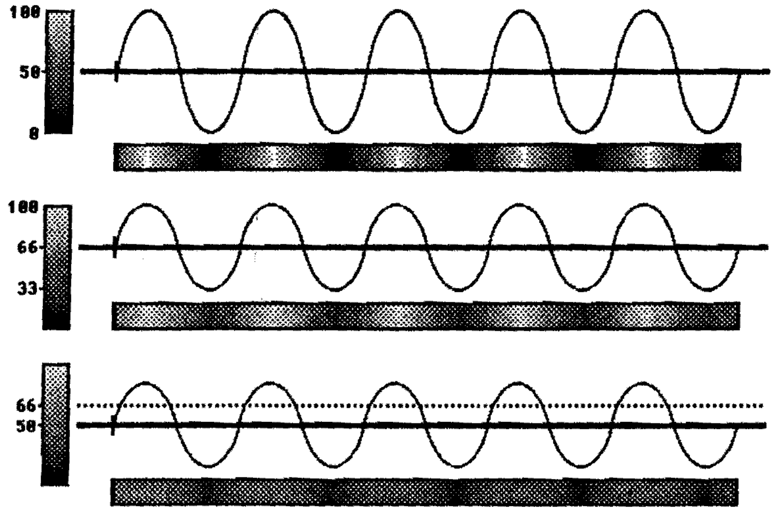


Figure 5.3: Sample wave patterns with varying degrees of clipping.

appear at a different gray level). The **Upper Clip (0-100)** and **Lower Clip (0-100)** input fields are provided so that you can specify a smaller constraint for the wave. See Figure 5.3 for a few examples of different clipping values.

The **Ramp (0-100)** and **Level Off (0-100)** input fields are measured as percentages. They perform the same function as in the operator, but are specified as values between 0% and 100%.

Once you are satisfied with all the ripple definitions, press the **Accept** button to create the rippled image. Press the **Cancel** button to return to the main screen without creating an image.

5.5 TEMP

The TEMP loader allows you to load the image currently in the MorphPlus temporary (undo) buffer into the primary image buffer. If no previous image was saved to the temporary buffer, then the image currently in MorphPlus will remain unchanged. For a discussion of saving to this temporary buffer, please read Section 6.4.

Besides using this buffer for "undo" operations, you can also use it for quick retrieval of the same image. One application of this technique would be to create various versions of the same image. You would save the image with

Format	Extension
ANIM	.anim
IFF	.brush
	.iff
	.ilbm

Table 5.1: A list of some of the file formats detectable by the UNIVERSAL loader and the file name extensions recognized as hints.

the TEMP saver, make the first version's changes, and save it out with one of the savers. Press the **Load** button (which loads the original image from the temporary buffer), make the second version's changes, and save those out. Continue pressing the **Load** button for subsequent versions.

5.6 UNIVERSAL

The UNIVERSAL loader attempts to identify which file format a specific file is written in. If successful, the appropriate loader will be called automatically. The UNIVERSAL loader makes loading images of different file formats a bit more convenient since you do not have to manually switch between loaders. The penalty for this convenience is a minimal slow down due to bringing two loaders in from disk instead of one. But, with a reasonable hard disk drive, this slow down in inconsequential.

The UNIVERSAL loader attempts to find conclusive evidence identifying which file format a given file is written in. However, some file formats do not provide any way of locating conclusive proof of their identity. If the UNIVERSAL loader cannot locate conclusive proof, it does contain heuristics which enable it to guess (correctly) at a file type most of the time.

The UNIVERSAL loader identifies file formats in a two step process. First, it uses file name extensions as a hint of which file formats to investigate first. Second, it checks the internal structure of the file against known facts about each file format. Table 5.1 lists some of the formats detectable by the UNIVERSAL loader including the file name extensions which it will recognize as hints.

If the UNIVERSAL loader cannot determine the file format for a given file, it will tell you. The UNIVERSAL loader will also inform you if it can determine which format a given file is written in, but you do not have a loader for that type.

Chapter 6

Savers

This chapter provides detailed descriptions of the use and operation of all savers which are included with MorphPlus. ARexx control of these savers is described in Section 12.4.

6.1 ANIM

The ANIM saver allows you to save the current rendered image data (either the cropped screen or entire size) as the last frame of an IFF-ANIM file or as the first frame of a newly-created ANIM.. Raw data (8 and 24 bit-plane IFF) is not currently supported in the IFF-ANIM file specifications.

After selecting the **ANIM** saver from the **Save Format** button and pressing the **Save** button, the ANIM saver control panel, as shown in Figure 6.1, will appear atop screen.

The first step you need to do is select the IFF-ANIM file onto which the current image data will be appended. Select the **Select File...** button. A file requester will appear to allow you to select the ANIM file. If you select a file that is a valid IFF-ANIM, the current rendered image data will be appended to it. If you enter the name of a non-existent file, the ANIM saver will create a new IFF-ANIM file, using the rendered data as the first frame of the animation.

The upper half of the control panel is where information about the selected ANIM file will be displayed. To display the number of frames in the animation, select the **Count Frames** button. Note that this process may take some time to complete, especially for ANIM files that contain a lot of frames. Once you select an ANIM file, its filename will appear to the right of the **File Name:**

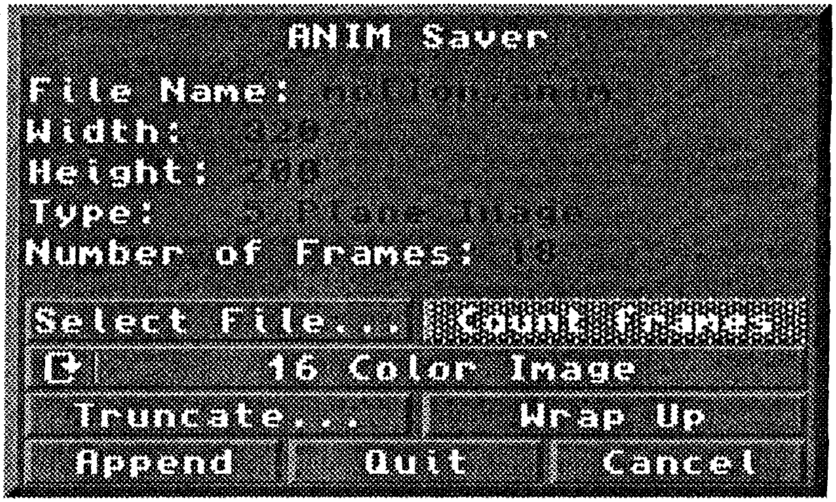


Figure 6.1: The ANIM Saver control panel. The **Count Frames** button has already been selected (**Number of Frames:** is not zero), causing it to be ghosted.

label and the width, height, and type of each frame will be displayed next to the **Width:**, **Height:**, and **Type:** labels. After retrieving the frame count, this button will become unselectable (the button's image will be ghosted).

As previously stated, ANIMs can only be made up of rendered image data. The type of rendered data can be selected from the **Save Type** cycle gadget. Clicking on this button will toggle between **Cropped Screen Image** and n **Color Image**, where n is the number of colors in the rendered data.

If the current rendered image data does not match all of the selected ANIM's attributes (width, height, and type), a message will appear. The rendered data must be the same as what the ANIM expects, otherwise this saver will not be able to append it to the file.

Not only does the ANIM saver allow you to append an image to an ANIM file, it allows you to modify an existing ANIM file on disk.

The **Truncate...** button allows you to lessen the ANIM to fewer frames than it currently has. Selecting this button will bring up another panel with a integer input field. You can either enter a positive or negative number. If the number is positive, the ANIM will be truncated to that many frames. If you enter a negative number, that many frames will be removed.

The **Wrap Up** button will copy the first and second frames of the animation to the end of the file. This is part of the IFF-ANIM file format specification and is used to produce a smooth looping animation. This operation will close the ANIM file. This means that the next time you enter this saver, you will have to select a new ANIM file to work on.

To actually append the current rendered image data, select the **Append** button. The meter window will appear, showing the progress of the save. If any errors occur, a message will be displayed. If no errors occurred, the control panel will disappear.

A time-saving feature of the ANIM saver is that it remembers the position of the last frame in the ANIM file. This allows you to append successive frames without having to wait for the saver to find the end of a file for every new frame. For animations made up of a few frames, this feature might not be as evident as with a "multi-frame" animation.

To exit the saver without actually appending the image, select **Cancel**. The control panel will disappear, returning control back to the main screen. If you want to use the time-saving feature just described, select this button to exit the control panel.

If you want to exit the control panel without saving the information about the last frame's position, select the **Quit** button. **Quit** will close the ANIM file, allowing you to use it in other programs.

NOTE: If you want to play the animation with an ANIM viewer, you MUST select the **Quit** *button after finishing all of your modifications with MorphPlus, select a different animation file using the* **Select File...** *button, select* **Wrap Up**, *or exit MorphPlus altogether. If you fail to do any of the steps, other programs will not be able to use the file.*

6.2 HAME

The HAME saver gives you control over the HAME display device from Black Belt Systems. The HAME works in two modes.

First, there is register mode. In register mode, the HAME displays an image with 2 to 256 palette mapped colors. To produce a register mode image for use with the HAME saver, simply produce a rendering having between 2 and 256 colors and invoke the HAME saver. Use of the enhanced palette mode is suggested.

Second, there is the HAME mode. In HAME mode, the HAME displays an image using the same sort of tricks that the Amiga's own HAM mode employs

to gain the impression of higher color content without greatly increasing the memory required to display such as image.

HAME mode is supported in both MorphPlus' HAM mode as well as MorphPlus' HAM8 mode. When used in conjunction with renderings performed in MorphPlus' HAM8 mode, HAME mode can show colors from a 262,144 color palette. In either case, the use of the enhanced palette mode is suggested.

Creating a HAME mode rendering in MorphPlus' HAM mode is quite simple. Just perform the following steps:

1. Select the HAM rendering mode in MorphPlus and render an image as you normally would.
2. Select the HAME saver and use it to either display or save an image in the HAME format.

Creating a HAME mode rendering in MorphPlus' HAM8 mode is slightly more complex. Follow these steps:

1. Enter the Palette control panel. Select **HAM8** as the **Color (Total)** value.
2. Select some number less than or equal to 60 for **Colors (Used)**. At 60 colors used, image quality will be maximized.
3. Ensure that **Offset Color Zero** is set to 0. You are encouraged to utilize the enhanced palette mode.
4. Accept these settings on the Palette control panel.
5. On MorphPlus' main control screen, set the rendering type to **CUST**. Also, select the desired screen settings, such as interlaced or overscan. Render the image by hitting the **Execute** button.
6. Select the HAME saver and use it to either display or save an image in the HAME format.

Upon entering the HAME saver, it automatically detects if the image data is rendered in a palette mapped mode (2 to 256 colors) or in a HAM mode (HAM or HAM8). If in a palette mapped mode, the HAME saver automatically reformats the rendered image to create a register mode image. If the rendered image data is in a HAM mode, the HAME saver automatically reformats the rendered image data into a HAME mode image.

The HAME saver will inform you which type of image it has created via a text message contained in the saver's control panel. The control panel gives you the choice of displaying or saving the HAME image. If you elect to display the image, it will be displayed until you click the left mouse button. Note that

unlike elsewhere in MorphPlus, you cannot scroll this display. If you elect to save the image, a file requester will appear for you to select the name under which the image should be saved.

The **Over** button on the HAME control panel allows you to set the type of overscan with which the image will be displayed. **Over W** will display only horizontal overscan. **Over H** will display only vertical overscan. Finally, **Over WH** will display both vertical and horizontal overscan. When the **Over** button is set to simply **Over**, the image will be displayed with no overscan at all.

NOTE: The HAME device is a low resolution device. That is, while the saver pays attention to the buttons defining overscan, interlace, and choosing between NTSC and PAL, it does not pay attention to the button choosing between high and low resolution.

6.3 IFF

The IFF saver allows you to save the current raw, rendered, or screen image data in IFF-ILBM format. You will be asked to select one of the following:

- **Cropped Screen Image** — Selecting this mode will save the screen image data which would be displayed if you select the **ReDisplay** button. This choice is available only if you have rendered in an Amiga displayable mode.
- *n* **Color Image** — Selecting this mode will save the entire rendered image. The number of colors in the rendered image will be presented in place of the *n*. This choice is available whenever rendered image data is available even if the rendering mode is not directly Amiga displayable.
- *n* **Bit-Plane Raw Image** — Selecting this mode will save the entire raw image as either a 24 bit-plane color image or 8 bit-plane grayscale image.

Whenever a raw image is saved, MorphPlus includes Commodore standard `CLUT` chunks which represent your current color balancing settings. However, many programs which now support Commodore standard 24 bit-plane IFF files do not yet support `CLUT` chunks. To enable you to get exactly the image you intend, you can use the Apply_Map operator to permanently modify the raw data to include the effect of the `CLUT` chunks.

Raw images saved in the IFF format by MorphPlus also include an `ASDG` chunk which preserves the internal representation of your current color balancing settings. Reading and rewriting such a file in a non-ASDG program will more than likely cause the `ASDG` chunk to be removed. However, this means only that

the color balance settings are lost and not that the data has been modified. You can reset the color balance settings by hand if you need them again.

After selecting which type of data to be saved by the IFF saver, the file requester will appear allowing you to specify the output file destination.

6.4 TEMP

The TEMP saver allows you to easily and quickly make a "backup" copy of the image currently in MorphPlus. It uses the image buffer reserved for use by MorphPlus (the size of which can be displayed by clicking on the **About** button on the main control screen).

The TEMP saver, in conjunction with the TEMP loader, will allow you to always have a backup copy of your work in progress. Before you make a change which might produce unknown or undesirable results, first save the image using the TEMP saver. If after processing the image you are not satisfied with the results, simply use the TEMP loader to reload the unmodified image. If the image came out the way you intended, then proceed with more changes or resave the new image with the TEMP saver and continue this step-wise process.

Although the TEMP saver can be used for "undo" operations, it can also be used as a way of setting up the use of the same image for manipulation without having to load the original image every single time. See the discussion of the TEMP loader for more details.

Chapter 7

Operators

This section provides detailed descriptions of the use and operation of all operators which are included with MorphPlus. ARexx control of these operators is described in Section 12.5.

7.1 Apply_Map

Changes made with the **Color Controls**, such as **Brightness** or **Gamma**, do not actually change any of the raw image data. Rather, these changes affect only look-up tables through which the raw image data is filtered. Therefore, the affect of changes made with the **Color Controls** can be undone.

The Apply_Map operator applies the changes made with the **Color Controls** to the actual raw image data. It then resets the **Color Controls** to a neutral setting. The Apply_Map operator, once started, cannot be aborted.

Three uses of this operator come to mind:

1. It allows the range of the **Color Controls** to be extended, by allowing you to repetitively apply color control changes over and over again.
2. It allows for better results when merging images using the image compositing feature of MorphPlus. For instance, you might want to merge an image with bright color settings with an image with dark color settings. Using this operator on both images prior to compositing will allow you to get the desired results.
3. It allows other Amiga programs which can read 24 bit-plane IFF images but do not parse "color look-up table chunks" to get the image data you

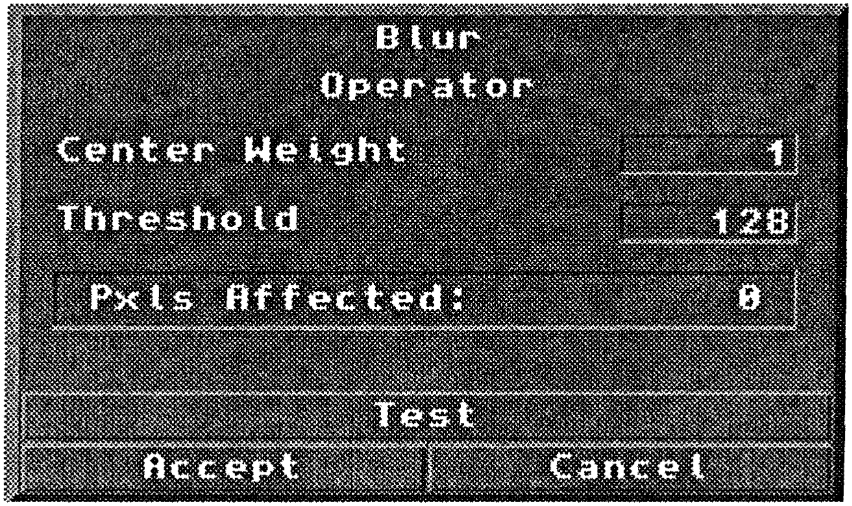


Figure 7.1: The Blur control panel.

intended.

7.2 Blur

Selecting the Blur operator causes the control panel shown in Figure 7.1 to be displayed. You can use this control panel to select or test the amount of blurring that will be done on an image.

The Blur operator allows you to blur certain pixels which differ from their neighbors by more than a user-definable threshold. Each pixel (weighted by a factor called the center weight) is averaged with the 8 surrounding pixels. If the resulting average differs from the actual pixel value by more than the user-defined threshold, the pixel is replaced by the average value. If the difference between the average and the original pixel does not exceed the threshold, the pixel is unchanged.

A center weighting factor from 0 to 16 may be specified. A lower center weight causes a more dramatic blur of the image.

A threshold of 0 to 255 may be specified. A value of zero forces all pixels to be blurred. A higher threshold will reduce the number of pixels which will be blurred.

The Blur operator allows a given set of values (for center weight and threshold) to be "pencil checked" using the **Test** button. By selecting this button, all computations will be made, but no changes will be made to the original data. The number of pixels which *would have been* affected will be reported. You can use this feature to fine tune your selections of center weight and threshold.

Selecting the **Cancel** button will exit from the operator without modifying the data. Selecting the **Accept** button will apply the currently defined blurring operation to the raw data.

7.3 Color_To_Gray

The Color_To_Gray operator provides three very important benefits beyond simply turning color images into grayscale images:

1. The method used within MorphPlus to convert from color into grayscale can actually produce grayscale data of considerably better quality than the original color data. For example, the Sharp JX-100 portable scanner is accurate in grayscale to 6 bit-planes and 18 bit-planes in color. However, when MorphPlus converts an 18 bit-plane color image into grayscale, it produces a grayscale image which is accurate to 8 bit-planes.
2. MorphPlus renders extremely well in grayscale due to its dithering capabilities and the fact that 16 grayscales cover the range of gray proportionately better than 16 colors cover the spectrum of all possible colors. For this reason, a high resolution interlaced gray image rendered with MorphPlus can look very nearly photographic, even given the Amiga's limited display capability.
3. This feature provides a way to convert color images into high quality grayscale images for the purpose of black and white desktop publishing. While color DTP is just now coming into its own, DTP is still predominantly grayscale.

Selecting the Color_To_Gray operator causes the control panel shown in Figure 7.2 to be displayed. You can use this control panel to select the weights with which the red, green, and blue components of the image will be factored into the color to grayscale conversion.

Two pre-programmed weighting schemes are provided. These can be enabled by selecting either the **Average Weights** or **Luminance Weights** button. By selecting the button marked **Average Weights**, each of the red, green, and blue components of the image will be given equal weight as the image is converted to grayscale. In essence, each gray pixel will be the average of the red, green, and blue color pixel components.

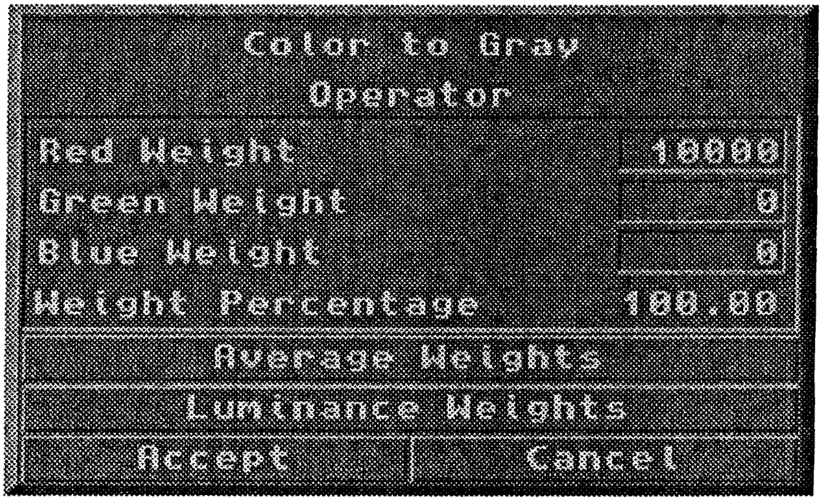


Figure 7.2: The Color_To_Gray control panel.

Selecting the button marked **Luminance Weights** will set up the weighting scheme used by the NTSC television standard to convert a color broadcast signal for display on a black-and-white television receiver.

If you wish to specify your own weights, you may do so by entering values into the three integer input fields supplied. The values you enter represent the fractional contribution that each color will have in the final averaging process. These values are scaled such that 0 represents 0 percent weight while 10000 represents a 100 percent weight. The total weighting is given below the three integer input fields and may exceed 100 percent.

In Figure 7.2, a weight of 10000 has been given to the red component while a weighting of zero has been given to the other colors. This will cause pixels with a large red component in the original will be shaded lightly in the resulting gray scale image. Pixels with very little red will be shaded darkly in the resulting gray scale image.

A total weighting percentage may exceed 100 percent. Pixels in the resulting grayscale image are clipped to the range of 0 to 255. Negative weights may also be specified.

Selecting the button marked **Accept** causes the currently defined conversion to take place. Selecting the button marked **Cancel** exits the operator without changing the current image in memory.

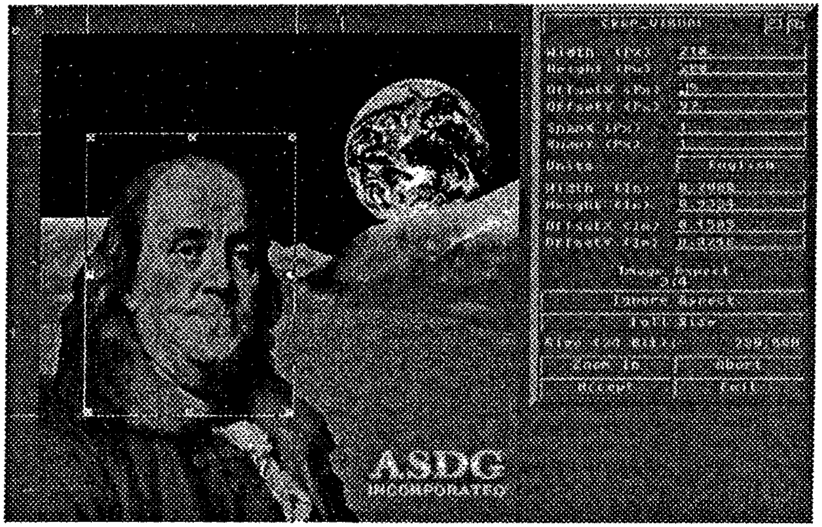


Figure 7.3: The Crop_Visual screen.

7.4 Crop_Visual

This operator provides an on-screen WYSIWYG user interface to the cropping operation.

The **Crop_Visual** control panel is shown in Figure 7.3.

The box representing the area of interest indicates where the cropping operation will take place. The portion of the image inside the box will be retained, while the area outside the box will be discarded. You can use the buttons marked **Full Size** and **Zoom In** to see finer or grosser detail. The **Accept** button will cause the cropping operation to actually take place. The **Cancel** button will leave the Crop_Visual operator without performing a crop.

Depressing `Shift-RETURN` (pressing the `RETURN` key while the `Shift` key is being held down) on your keyboard is the same as hitting the **Accept** button. When the control panel is zipped, hitting the **z** key on your keyboard is the same as hitting the **Zoom In** button. Also, when the window is zipped, hitting the **u** key on your keyboard will unzip the window, making it full size again.

7.5 DCTV

The DCTV operator reformats the raw image data currently loaded in MorphPlus' memory into the format used by the DCTV display device (from Digital Creations). The reformatting overwrites the rendered image area in MorphPlus' memory but does not disturb the raw image area. If MorphPlus is not already set for 8 colors, the operator automatically shifts MorphPlus into Hi Res, 16 colors as required for DCTV. After the operator completes, it will display the resulting DCTV formatted image. This image cannot be scrolled from within MorphPlus.

If MorphPlus was already set to 8 colors, a 3 bit-plane DCTV image will be created instead of a 4 bit-plane version.

The DCTV operator takes its display size and type (i.e., interlace on or not, PAL or NTSC, etc.) from MorphPlus' current screen control settings.

You can save the DCTV version of the image by selecting the IFF saver. When saving a DCTV image, make sure you select the button marked *n* **Color Image** where *n* is either 8 or 16.

NOTE: This operator requires and uses the `dctv.library` *available from Digital Creations. If this library is not present in your* `LIBS:` *directory, you will get a message indicating a "fatal error in the operator".*

7.6 Define_Pxl_Aspect

MorphPlus adjusts the pixel aspect of the image as the image is scaled. After many scaling functions, the pixel aspect of the image may have little information left in it. For example, after scaling an image several times, you might arrive at an image that you wish to use as a low resolution interlaced picture. You don't care what the pixel aspect was before you started, but now that you have a picture you want to keep, you might want to redefine the pixel aspect to be exactly right for low resolution interlaced.

Also, many applications (such as desktop publishing) can make use of other resolution and sizing information which some file formats can maintain.

Using the Define_Pxl_Aspect operator, you can redefine the pixel aspect of the currently loaded image to be whatever value you wish. You can also redefine the "resolution" of the image so that programs such as desktop publishing packages will believe your image should be whatever size you want it to be.

Figure 7.4 shows the control panel for this operator. It displays the current

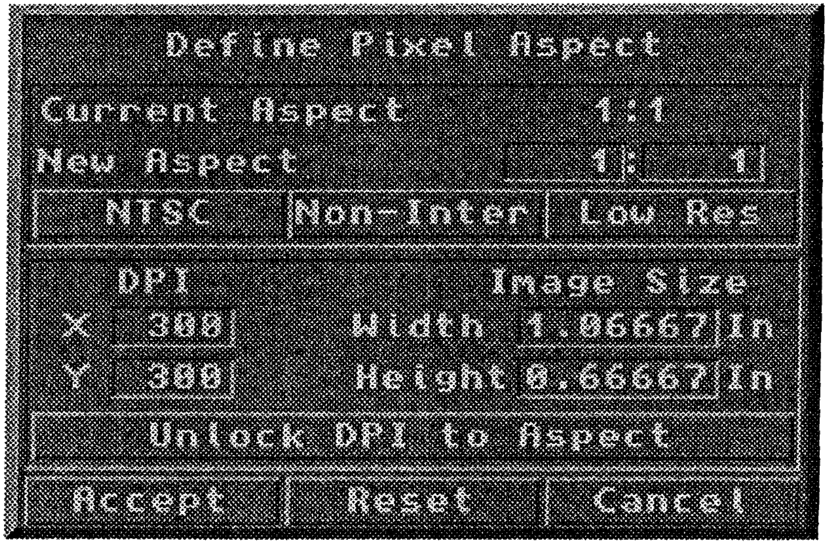


Figure 7.4: The Define_Pxl_Aspect control panel.

pixel aspect and has two integer input fields which allow you to enter a new pixel aspect numerically. You can also specify a new pixel aspect by selecting the desired screen parameters from the buttons located just below these input fields.

Further, you can define an x and y resolution for your image. These values are maintained by MorphPlus and are actually used (in creating the rulers) in the operators.

Note that this operator does not actually modify any of the image data. It simply redefines MorphPlus' internal idea of the image's pixel aspect and resolution. The new pixel aspect and resolution information will be saved with the image when the image is saved in a format which supports such information.

7.7 Gray_To_Color

The Gray_To_Color operator provides the logical inverse of the Color_To_Gray operator. Gray_To_Color reformats the internal data representation of raw grayscale image data into that of raw color image data. This feature is provided so that you can perform all image processing functions on both color and grayscale images.

When combined with image compositing, interesting artistic effects can be created. For example, a masked color image can be "pasted" into an otherwise completely grayscale image for a startling effect. See **Merging a Color Image Into a Grayscale Image** in Appendix C for more information.

7.8 Horizontal_Flip

Many applications require the production of mirror images of the original data. Specifically, this is required by screen and heat-transfer printers. Selecting the Horizontal_Flip operator will horizontally flip the raw image data currently in memory.

Note that performing a horizontal flip followed by a vertical flip is the same as rotating the image 180 degrees. This can be used in conjunction with the **Orientation (Port / Land)** button to rotate images through 0, 90, 180, and 270 degrees. Note, as well, that you can also do these same rotations with either the Rotate or Perspective operators.

7.9 KillTemp

The KillTemp operator allows you to clear the image currently in MorphPlus' temporary buffer (placed there by the TEMP saver), whether or not an image exists in the buffer.

7.10 Perspective

With the Perspective operator, you can not only scale the image (similar to the Scale operator), but also reorient its position and the angle at which you look at it. Think of the image as floating in 3-dimensional space and you, the camera, able to move around in this space. You are able to see any or all of the image and look at it in any orientation you desire. The ability to do this is possible with this operator.

Select the Perspective operator from the MorphPlus operator list. Please refer to Section 2.4.1 for specific instructions.

After selecting the operator, the Perspective screen (titled **Perspective_Visual** should appear with up to four windows—titled Preview, Camera, Rotation, and Output—on it. The Perspective control panel window contains all of the controls available for manipulating your image. The Preview window

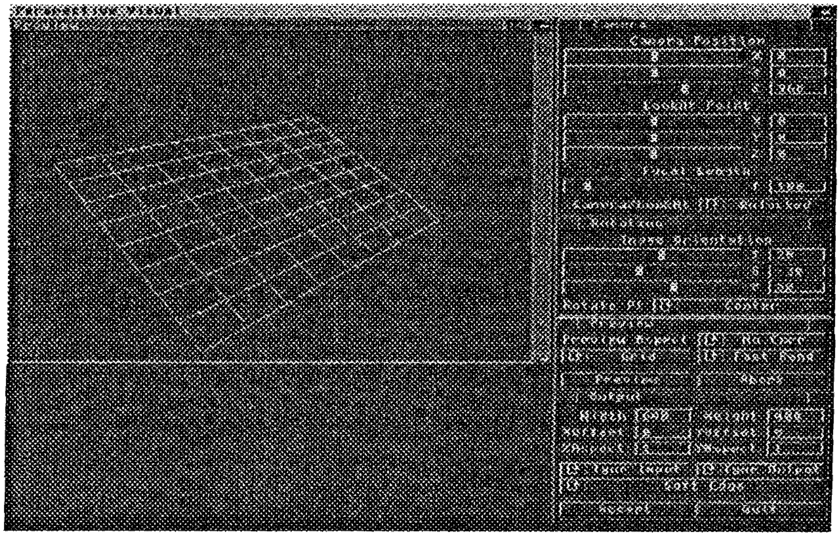


Figure 7.5: The Perspective Operator screen.

is where you can see a visual representation of the current perspective settings, as well as a preview of the modified image using these current settings.

7.10.1 Camera Control Panel

The Camera control panel can be displayed by selecting the **Open...** menu item in the **Camera** menu. To close it, either select the **Close** menu item (in the same menu) or click on the window's close gadget.

Your (the camera's) position, relative to the image, in 3-dimensional (3D) space is controlled by the **Camera Position** controls. The (x,y,z) location is set with the **X**, **Y**, and **Z** value sliders and input fields, respectively. By changing these values, the camera can be repositioned.

The location in 3D space at which the camera is looking is described by the **LookAt Point** values. Again, this location is specified as (x,y,z) using the **X**, **Y**, and **Z** value sliders and input fields. Note that you do not have to always look at the center of the image. The **LookAt Point** has the same "freedom" of movement as the **Camera Position**.

The fractional distance between any two adjacent points in 3D-space (for example, points (1,1,1) and (1,1,2)) can be described by the focal length (**f**) value slider and input field. Under normal cases, the distance between two adjacent points is 100 percent of a unit length. This would be shown as an **f** value of

100. If you want each unit to actually be 50% of what it should be, set **f** to 50. This **f** value causes distances to seem closer together. You can also set **f** to percentages larger than 100% to make distances seem farther apart.

Sometimes you would like to lock the distance between the **Camera Position** and **LookAt Point**. This can be done by setting the **Camera/LookAt** button to the **Locked** position. When locked, any change to the **Camera Position's X, Y,** or **Z** values causes an equal change to the corresponding **LookAt Point X, Y,** or **Z** value.

Take a look at Figure 7.6 for examples of different Camera/LookAt values.

To reset only the above values to the values that were shown at the time this operator was selected, select the **Reset** menu item in the **Camera** menu.

7.10.2 Rotation Control Panel

The Rotation control panel can be displayed by selecting the **Open...** menu item in the **Rotation** menu. To close it, either select the **Close** menu item (in the same menu) or click on the window's close gadget.

You can control the image's orientation with the **Image Orientation** value sliders and input fields. The **T** (theta, or θ) control defines amount of rotation about the axis that lies directly on the image plane's surface normal vector, with positive **T** values rotating the image in a counter-clockwise direction around this axis. The **G** (gamma, or γ) control defines the amount of rotation about the absolute y axis, with positive **G** values rotating the image in a clockwise direction. The **P** (phi, or ϕ) control defines the amount of rotation about the horizontal axis (affected by the **G** value), with positive **P** values rotating the image in a clockwise direction.

The point on the image which the rotation will move around is defined by the **Rotate Pt** value. Click on this cycle gadget to scroll through the available choices. The **Center, Top Left, Top Middle, Top Right, Right Center, Bottom Right, Bottom Middle, Bottom Left,** or **Left Center** point on the image can be used, as well as the currently defined **LookAt Point** (defined in the Camera window).

Figure 7.6 has an example (the skewed image) of changes to all three **Image Orientation** values.

To reset only the above values to the values that were shown at the time this operator was selected, select the **Reset** menu item in the **Rotation** menu.

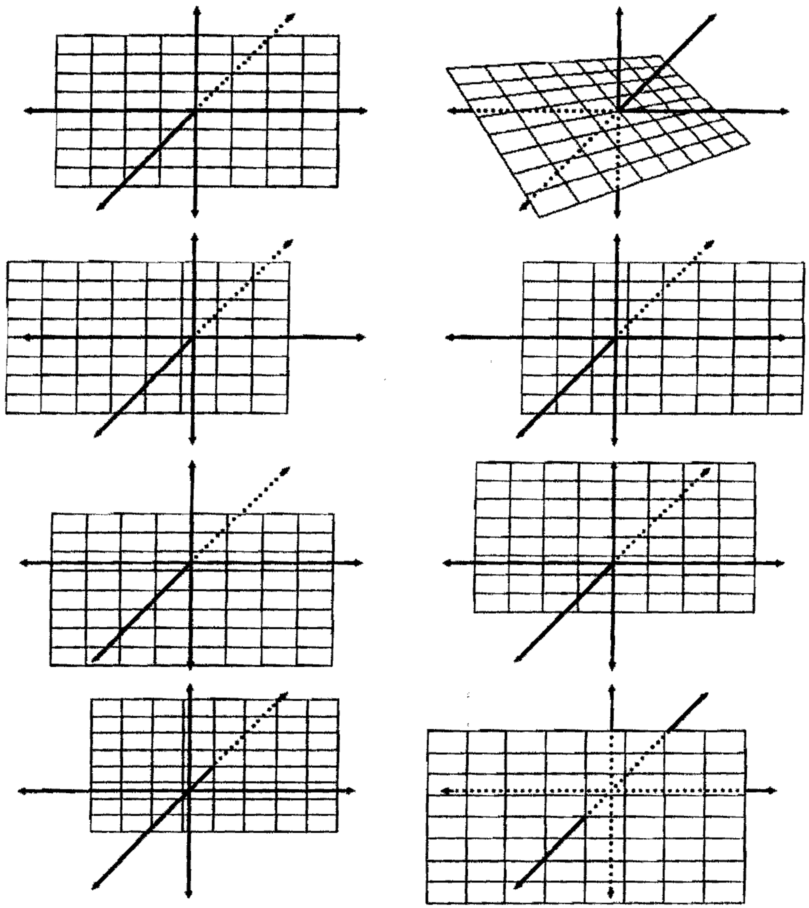


Figure 7.6: Sample image orientations. Going from left to right, top to bottom: normal, skewed, negative x, positive x, negative y, positive y, negative z, and positive z.

7.10.3 Preview Control Panel

The Preview control panel can be displayed by selecting the **Open...** menu item in the **Preview** menu. To close it, either select the **Close** menu item (in the same menu) or click on the window's close gadget.

The image in the Preview window can be represented in "wireframe" in one of three ways: as a gridded rectangle (**Grid**), a rectangle with the word "ASDG" on it (**ASDG**), or as a plain or empty rectangle (**Rect**). The **Grid** and **ASDG** rectangles are very helpful for visualizing the orientation of the image. Note that the **ASDG** representation is not a shameless advertising plug on our part—it does have a valid use. For example, if you chose a **Camera Position** with negative **Z**), you wouldn't be able to tell, without the use of some text on the rectangle, just what side (front or back) of the image you were actually looking at. The word "ASDG" just happened to be as good of a word as any other.

The pixel aspect of the preview rendering (shown in the **Preview Aspect** cycle gadget) can be specified by selecting **No Corr**, **1:1 Corr**, or **Output Corr**. **No Corr** will not correct for the pixel aspect. **1:1 Corr** will display the image as if it were being displayed on an output device that has a 1:1 pixel aspect. **Output Corr** will display it using the currently defined Output pixel aspect (**XAspect** and **YAspect** in the Output control panel).

The quality of the preview rendering can be controlled by selecting either **Good Rend** or **Fast Rend**. **Good Rend** produces a better rendering but is slower, whereas **Fast Rend** is faster but doesn't render as well. If you want to get a rough idea of the current perspective settings, use **Fast Rend**.

To preview the current settings, select the **Preview** button. The image that would be produced with these settings is shown in the Preview window. To abort the preview rendering at any time, select the **Abort** button.

To reset only the above values to the values that were shown at the time this operator was selected, select the **Reset** menu item in the **Preview** menu.

7.10.4 Preview Window

The Preview window displays the image as it would look if the current settings, defined in the Camera, Rotation, Preview, and Output control panels, were used.

The width and height of the resulting image (which may be larger than the area occupied by the original image) is represented by the inner border of the window. The window can be shrunk or enlarged, but it will always preserve the relationships among the **Width**, **Height**, **XOffset**, **YOffset**, **XAspect**,

and **YAspect**. This will cause the window to adjust itself to the proper size, even though you resized it differently.

7.10.5 Output Control Panel

The Output control panel can be displayed by selecting the **Open...** menu item in the **Output** menu. To close it, either select the **Close** menu item (in the same menu) or click on the window's close gadget.

The width of the resulting image can be set by changing the value in the **Width** input field. The height is set in the **Height** input field. These two values let you modify the size of (or scale) the image, which most operators do not let you do. Changing either of these two values will automatically adjust the Preview window's size to properly represent the new size.

The offset of the actual image from the left edge of the final picture is controlled by the value in the **XOffset** input field. The top edge offset is controlled by the **YOffset** value.

The rendered preview can be displayed using a user-defined pixel aspect by entering values in the **XAspect** and **YAspect** input fields. This can be very handy for when you want the rendering in the Preview window to match the pixel aspect of a given output device.

The pixel aspect, as defined in MorphPlus and modifiable with the Define_Pxl_Aspect operator, can either be ignored or corrected for by selecting either **Ignr Input** or **Corr Input**, respectively. **Ignr Input** will not use the MorphPlus-defined pixel aspect, but instead will assume a 1:1 aspect. **Corr Input** will take the current pixel aspect value into account when rendering the preview.

The pixel aspect used in displaying the Preview window image can similarly be controlled by selecting either **Ignr Output** or **Corr Output**. **Ignr Output** will assume a 1:1 output device, whereas **Corr Output** will assume a **XAspect:YAspect** output device.

If you want to scale your image so that it will look correct on an output device with a specific pixel aspect, make sure that the input and output aspect values are corrected (**Corr Input** and **Corr Output**), and that the **XAspect** and **YAspect** values match your output device's pixel aspect.

The edges of the image may be anti-aliased by setting the **Edge Type** to **Soft Edge**. **Hard Edge** will not anti-alias the edges. **Soft Edge**'s main use is to create grayscale images for use as alpha channels in programs, such as Art Department Professional, that support them. These alpha channels can be created with **Soft Edge** to smooth out the aliased edges that a rotated or skewed image (generated by the Perspective operator) may produce. These

alpha channels are great for compositing one image over another, without the "hard edged" look that can result.

To reset only the above values to the values that were shown at the time this operator was selected, select the **Reset** menu item in the **Output** menu.

If you are satisfied with the current settings and want to actually modify your image, select the **Accept** button or menu item (in the **Project** menu). To return to the MorphPlus screen without modifying the image, select **Quit** (either button or menu item).

7.10.6 Other Controls

Although you can selectively reset the attributes in any of the panels, to reset all attributes to their default or last saved values, use the controls in the **Settings** menu. **Reset to Default** resets all controls to their default, never-changed, values; **Last Saved** resets them to the values they had the last time the operator was used, and hence, the values at the time the control screen and panels were opened (for the current session).

7.11 Refract

The Refract operator allows you to add wave-like patterns to your image. This effect is similar to the Ripple operator's results, except that you are not restricted to just waves of concentric circles—the actual wave patterns are defined by the values in a grayscale image.

Don't let the simplicity of the Refract control panel (shown in Figure 7.7) fool you...this operator can create very startling, and controllable, effects. By entering values into the **Pool Depth** and **Refraction Index** input fields, you can change the shape and behavior of the body of water on which these rippled patterns will appear.

Think of your image as being at the bottom of a pool of a particular medium (for the Ripple loader and operator, it is water). The **Pool Depth** value specifies the depth (in pixels) of this pool, from the wave-like pattern at the top to the image at the bottom. Changing the depth affects the way the image appears through the rippled patterns.

The **Refraction Index** value describes the type of medium (water, air, etc.) that is in our "pool." This index is a measure of how light rays are "bent" as they pass from one medium through another.

For a real-world pool, the light rays (from the sun) pass through one medium (air) through to another (water). This change of medium causes a change in

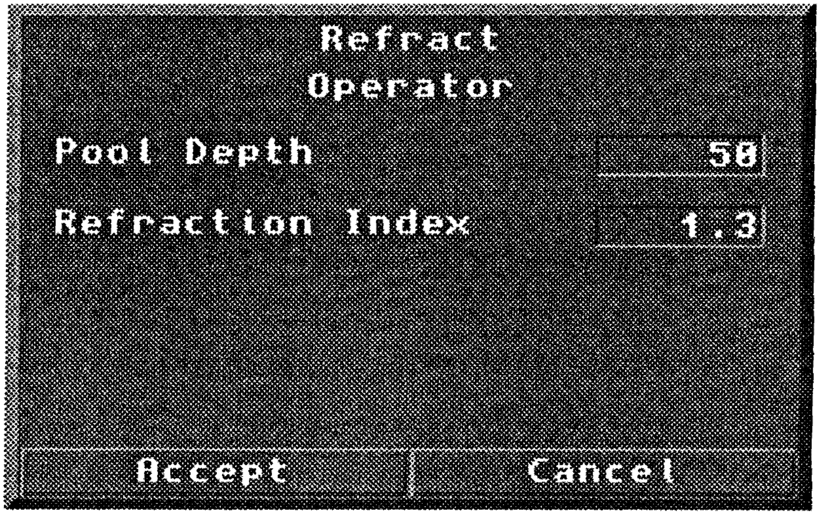


Figure 7.7: The Refract Operator screen.

Medium	Refraction Index
Air	1.00
Water	1.33
Glass	1.52

Table 7.1: Approximate indexes of refraction for various mediums.

direction of these light rays. If you were looking at the bottom of the pool from above (with a representation of your image somehow painted on the bottom of the pool), the image would become slightly distorted as waves start to appear on the water's surface.

The shape of these waves is described by the levels of gray in the grayscale IFF-ILBM image required by this operator. Just as with the RIPPLE loader, whiter areas of the image correspond to wave crests, while darker areas correspond to troughs (low points). It is through this representation of the medium's surface, along with the medium's characteristics (depth and degree of refraction) that the image will be modified. Table 7.1 lists a few indexes of refraction. You are encouraged to modify this value when experimenting with this operator.

If you press the **Accept** button, a file requester will appear from which you can select the IFF-ILBM image to use as the shape of the wave patterns. Grayscale

image data (in the same dimension as the image you are modifying) is required, although if you select a color image, it will be converted to gray.

To abort this operation, press the **Cancel** button.

A simple example of this operator's effect on an image can be seen in Appendix A.

7.12 Rendered_To_Raw

The **Rendered_To_Raw** operator converts the rendered data currently held within MorphPlus' memory back into the raw data it represents. In doing so, it replaces the raw data which was previously held in MorphPlus' memory. This operator makes it possible, for example, to save rendered data in file formats which only support raw data. Or, to send rendered data to devices whose savers only support raw data.

The operator can also be used for special effects. For example, if you wish to composite a true color object over a 16 color backdrop, you can render the 16 color backdrop, invoke this operator to convert the rendered 16 color data back into 24 bit-plane raw data, then composite in the true color foreground object.

This operator can fail only if there is not enough memory available to hold the raw version of the rendered memory currently in MorphPlus' memory.

7.13 Ripple

With the Ripple operator, you can produce a ripple-type effect on your image. A real world example of this is the pattern on the water's surface after you throw a pebble in the water. Concentric waves (waves sharing a common center) of rippled water, moving away from the center (where the pebble hit the water) would be produced.

Another way to visualize the effect on an image is to imagine that your image was somehow painted on the surface of water. Any slight movement would cause disturbance of the water's surface. Throwing in a pebble would cause waves to start moving away from the center of the point of impact. This operator will allow you to create this type of effect.

Unlike other programs that let you produce this rippling effect on your images, MorphPlus' Ripple operator lets you specify more than one ripple at a time. This allows you to create interference patterns between any two or more ripples (at those points where the waves of one ripple would come in contact with other

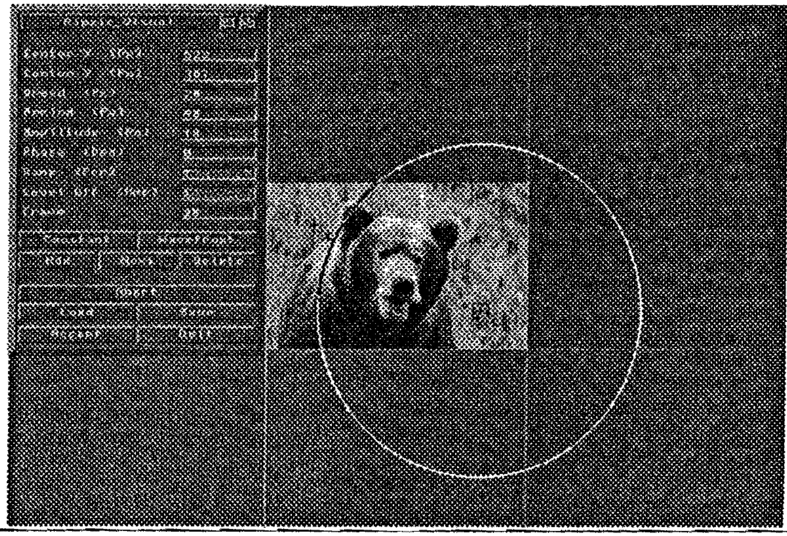


Figure 7.8: The Ripple Operator screen.

ripples' waves). Note that these interference patterns cannot be produced correctly if you create one ripple then place another ripple next to it (in a second pass).

Select the Ripple operator from the MorphPlus operator list. Please refer to Section 2.4.1 for specific instructions.

After selecting the operator, a preview screen should appear with the Ripple operator control panel on top of a grayscale screen. Within this screen is a grid of nine rectangles (outlined in white), with a grayscale representation of the current image in the center rectangle.

Before we proceed, we should clarify the usage of a couple of terms in this section. "Ripple" will describe each individual wave pattern definition. "Wave" will describe the individual crests (and corresponding troughs) of a ripple. Zero or more ripples can be defined for the image. Zero or more waves will be produced for each given ripple.

A Ripple Crosshair (as shown in Figure 7.9) visually describes the center or starting point of the ripple. As you can see in this figure, the Ripple Crosshair is made up of the crosshair itself, along with a letter (**C**, **I**, or **D**) to the lower right of it. The currently selected crosshair (as shown in the right side image in the above picture) has a small box encompassing it, whereas the unselected ones (left side image) don't. The letter next to each crosshair describes the change in a ripple's amplitude from the ripple's wavefront (the outer wave in

Figure 7.9: Representations of unselected and selected ripple crosshairs.

the series of waves) toward the center of the ripple. A **C** signifies Constant amplitude (no decay), **I** for Increasing amplitude, and **D** for Decreasing. This term will be discussed in more detail later in this subsection.

Although you can enter values into the control panel's input fields, the more intuitive way of specifying the location of the ripple's center is to move the crosshair. To do this, place the mouse pointer above the crosshair, hold down the left mouse button, move the mouse so that the crosshair moves to the new location, and release the hold of the mouse button. Notice that when you move it around the image, the values in the **Center X (Px)** and **Center Y (Px)** input fields change as well. This gives you real-time feedback of your changes.

To center the Ripple Crosshair horizontally on your image, select the **Center X** button. To center it vertically, select the **Center Y** button..

Note that you are not restricted to locations on the image itself. This is the purpose of using a grid (with your image in the center) for this operator. You can define the center of the ripple anywhere outside of the image. This will allow you to produce a section of the ripple across your entire image, which can be good for those times when you want a more straight looking wave (as opposed to a circular or arced one).

NOTE: Through the visual user interface (VUI) you might not be able to move a ripple's center far enough away from the image to produce these straight looking waves. If you need to create these types of waves, you should use the operator's ARexx interface which allows for x and y coordinates between -30000 and 30000.

Now that you've selected the center of your waves, you will want to specify the ripple's characteristics. These include its speed, period, amplitude, phase, and instance. Before we define these terms, let's discuss what this operator is trying to produce. A ripple describes the movement of a liquid surface over time, whereas a picture is a single image with no movement at all nor

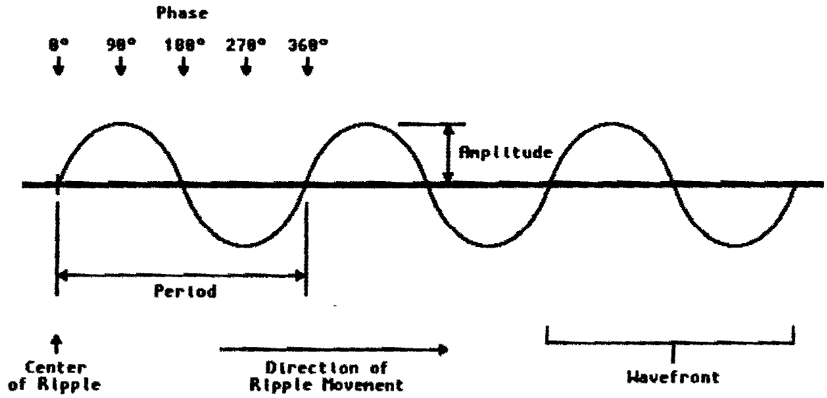


Figure 7.10: A ripple's characteristics.

characteristic of time. So how do we then define a ripple on an image? The answer is to think of a rippled image as a "snapshot" (a single instance) of that ripple's movement. When we describe the ripple's movement, we can use characteristics of moving objects (such as an ocean wave), but specify the instance in time (a unit of time is equal to one frame) we want to "look at".

The first characteristics we will look at is the speed of a wave. The speed can best be described as how fast (in number of pixels per frame) the wave will propagate from the "point of impact" (center of the waves). This value is specified in the **Speed (Px/Frame)** input field. For example, if the **Speed (Px/Frame)** was set to 10, at one instance (frame 1, for example), the wavefront (outer extent of the series of waves) will be 10 pixels away from the center. At the next instance (frame 2), that same waves will be 20 pixels away.

The period of a wave is the distance (in number of pixels) between two waves' crests, troughs, or other corresponding points. The higher (longer) the period, the farther it will be between similar points on the wave, and vice versa, for a given speed. This value is specified in the **Period (Px)** input field.

The height of a wave is called its amplitude. This is the distance between the midpoint on the wave and the highest point (the crest). It is also equal to half the distance between crest and trough. This value is specified in the **Amplitude (Px)** input field. The amplitude somewhat corresponds to the amount of distortion that will be seen. For example, a low amplitude will not distort the image (underneath the layer of "water") as much as a high amplitude, for a given period.

The phase of the wave describes the part of the wave that will start the wave at the ripple center. You can enter a value between 0 and 360 in the **Phase(Deg)**

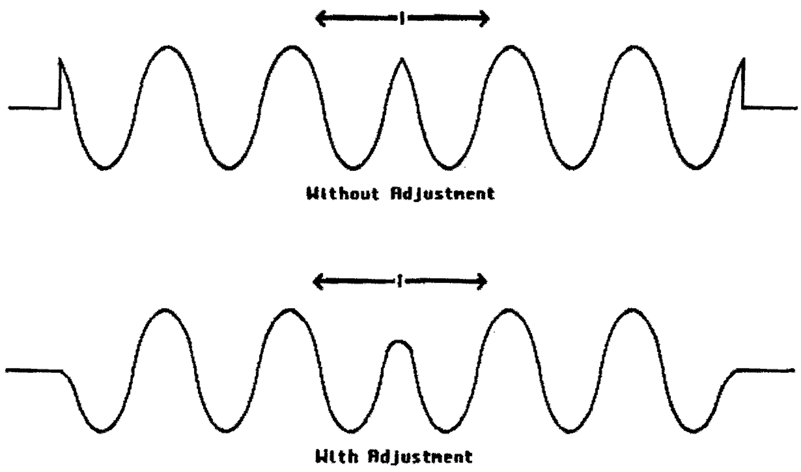


Figure 7.11: Adjustment of a ripple's waves to avoid unwanted results.

input field. This value corresponds to the position on the wave (a sinusiodal wave). See Figure 7.10 for a more concise description of the phase of a wave.

Note that if you specify a **Phase(Deg)** value that is neither 90 nor 270 degrees, the wave at the ripple center will not look smooth. Remember that a wave propagates in all directions, which might cause a pointed cone-looking crest or trough at the center. Real waves would not produce this result—the wave should not have "sharp" edges. The Ripple operator will smooth out this cone shape so that the wave will model a "real-world" wave more closely. Also, depending upon the **Phase(Deg)** value, the wavefront might also cause a similar, weird result. Luckily, the wavefront is also adjusted so that the wave flattens out at the points where it touches non-affected areas. See Figure 7.11 for an example of what is being done.

If you could freeze the motion of a ripple, you are looking at an instance of it. This instance (or frame, as in frame number) is selected in the **Frame** input field. At frame 1, the ripple is shown with no "motion" (the shape of the ripple at the beginning, or time zero). At frame 2, the ripple has moved away from the center a defined distance (specified by the **Speed (Px/Frame)** value). The higher the frame number, the farther the wavefront (the outermost wave) will be from the center.

Let's discuss the relationship between **Speed (Px/Frame)**, **Period (Px)**, and **Frame**. The number of waves that you will see at frame 1 (the first instance of the ripple) is defined by the period. If the speed is less than the

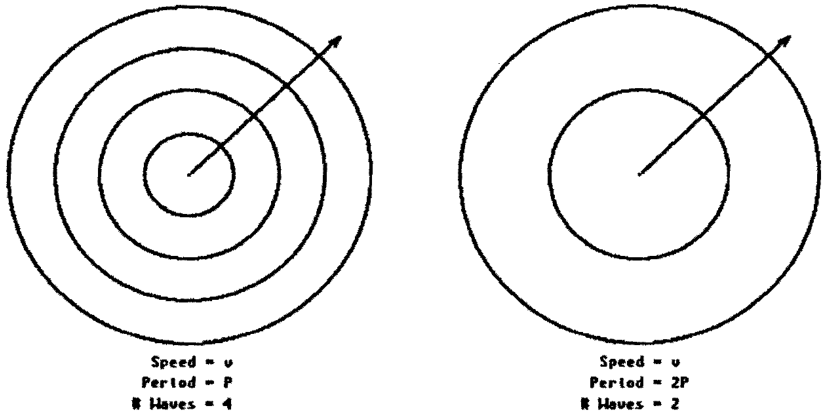


Figure 7.12: Comparison of speed versus period. Each circle represents one complete period.

period, each successive frame will only show a little more of the same portion of a ripple's "period" since the wave is not moving fast enough in each frame to produce a complete period. If the speed is greater than the period, then each successive frame will introduce more than one period portion of the wave since the ripple is moving fast enough in each frame to show more waves.

If you look at Figure 7.12, you will see two ripples with equal speed. The ripple on the left has a period (P) that is half as long (in pixels) as the ripple on the right ($2P$). As you can see, for a given speed, the longer the period the fewer the number of waves at a given instance (frame), and vice versa.

Each wave's amplitude, and hence its amount of distortion, over time can be controlled by choosing **Constant**, **Increase**, or **Decrease**. **Constant** keeps the amplitude of the waves the same across successive waves. **Increase** increases the amplitude as the waves get closer to the center of the ripple center; remember that the "older" portions of the ripple are at the wavefront (the outermost wave) and "newer" ones are closer to the center. The amplitude at which the increasing wave will level off into a constant amplitude can be specified in the **Level Off (Amp)** input field. **Decrease** decreases the amplitude over time until it reaches the amplitude defined in **Level Off (Amp)**.

For **Increase** and **Decrease**, you can control how many pixels the amplitude of each successive wave (from the wavefront) will increase or decrease; this value is specified in the **Ramp (Px/Period)** input field. This length, specified in the **Ramp (Px/Period)** input field, will be added (for **Increase**) or subtracted (**Decrease**) from the waves' amplitude until the value in **Level Off (Amp)** is reached. See Figure 7.13 for a comparison of the three wave behaviors.

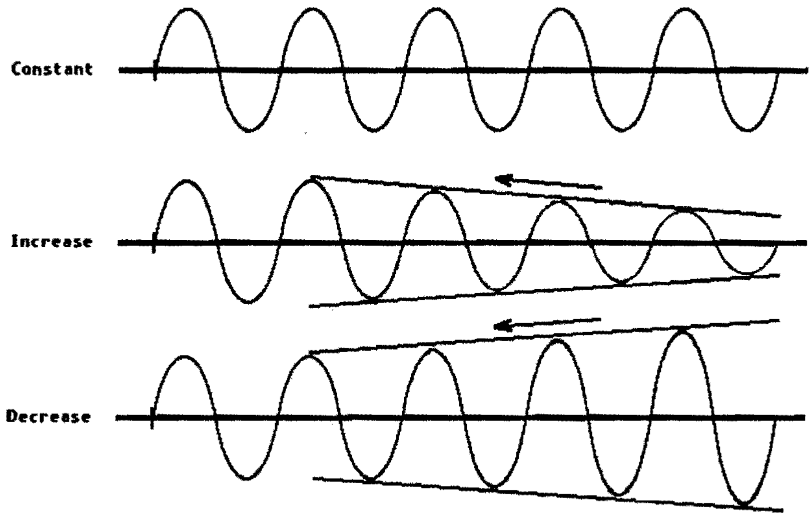


Figure 7.13: A comparison of the three amplitude behaviors.

If the **Wavefront Display Mode** button displays **No Wavefront**, only the Ripple Crosshair will be displayed. To help you visualize a ripple at the defined **Frame** number, you can choose to display its wavefront with a circle by clicking once on the **No Wavefront** button so that it will display as **Wavefront.**

If it displays **Wavefront**, for **Constant** amplitude, the location of the wavefront will be displayed as a circle. For waves that **Increase** or **Decrease**, the wavefront *and* the "level-off" amplitude (specified in the **Level Off (Amp)** input field) will be drawn.

As mentioned earlier, you can specify more than one ripple at the same time. This is done with the **Next**, **Add**, **Delete**, **Load**, and **Save** buttons. The **Next** button allows you to activate the next ripple definition on the screen. The active definition is denoted by a thin square around the active crosshair. **Add** adds another ripple definition; the added definition becomes the active one. **Delete** will remove the currently active definition and activate one of the previous definitions. Note that at least one vector must be defined at all times, which means that you will not be able to delete the last visible vector. **Load** allows you to load in a file of previously-saved set of ripple definitions, while **Save** lets you save the current set of definitions to a disk file. For both **Load** and **Save**, a file requester will be displayed from which you can select the desired file.

To accept the current set of ripple definitions and start using them to create the ripples on your image, select the **Accept** button. If you want to return

to MorphPlus' screen, select the **Quit** button. Note that when you exit the Ripple operator, your previously defined set of ripples are remembered. This allows you to "pick up where you left off" whenever you use the operator again.

If you load in a new image (with a size different than the previous picture) between invocations, any ripple centers that you have defined will be in the same screen position. This might move a ripple center inside or outside of the image, depending upon the width and height difference between the two images.

7.14 Rotate

With the Rotate operator, you can rotate circular areas of your image. The direction, center, amount, and quality of rotation, as well as the amount of blurring used at the edge of the circle, can be controlled.

NOTE: If you need to rotate the entire *image but also take into account your output device's pixel aspect, use the Perspective operator. The Rotate operator cannot properly adjust for a pixel aspect different than what MorphPlus believes this image's has.*

Select the Rotate operator from the MorphPlus operator list. Please refer to Section 2.4.1 for specific instructions.

After selecting the operator, a preview screen should appear with the Rotate operator control panel on top of a grayscale representation of the currently loaded image.

The Rotate Circle visually describes the circular portion of your image that will be rotated. As you can see in Figure 7.15, the Rotate Circle is made up of a thin circle with a crosshair at its center, a horizontal radius line in the right half of the circle, a small vertical line on this radius line, and a dashed line extended from the center of the circle.

Although values can be entered into the control panel's input fields, the more intuitive way of specifying the location of the rotate area on the image is to move the center crosshair on the Rotate Circle. Moving this crosshair moves the Circle as a whole around the image. To do this, place the mouse pointer above the crosshair, hold down the left mouse button, move the mouse so that the crosshair moves to the new location, and release the hold of the mouse button. Notice that when you move it around the image, the values in the **Center X (Px)** and **Center Y (Px)** input fields change as well. This gives you real-time feedback of your changes.

To center the Twirl Circle horizontally on your image, select the **Center X**

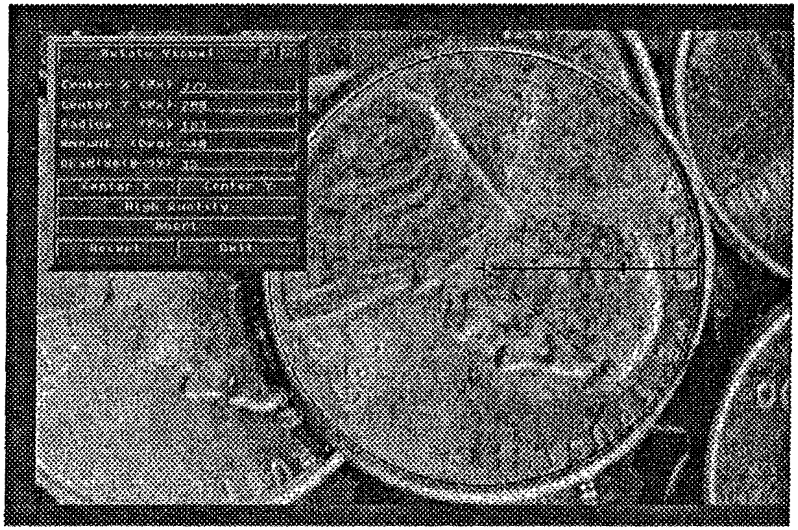


Figure 7.14: The Rotate Operator screen.

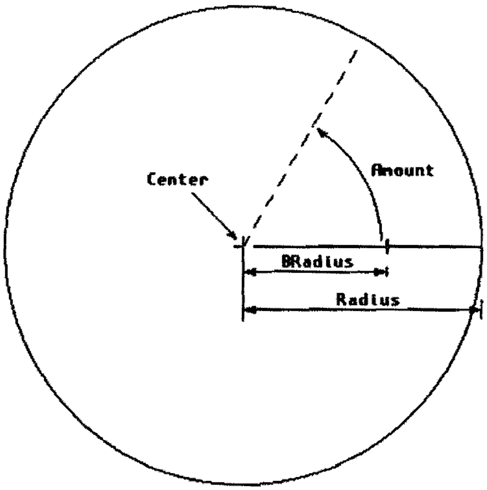


Figure 7.15: Components of the Rotate Circle.

button. To center it vertically, select the **Center Y** button.

Now that you've selected the center of your twirl area, you will want to specify the radius of this Circle. If you click the left mouse button while the pointer is anywhere but on the crosshair, the radius of the circle will snap to the new location. The actual radius (in pixels) is shown in the panel's **Radius (Px)** input field.

To define how much rotation will be done, you will have to enter a value in the **Amount (Deg)** input field. The current amount of twirl is shown on the Rotate Circle as a dashed line extending from the center of the Circle. An **Amount (Deg)** of 0 (no rotation) places the dashed line directly on top of the horizontal radius line. Positive **Amount (Deg)**s move the dashed line in a counter-clockwise direction. Negative **Amount (Deg)**s move it clockwise. Note that you cannot modify this dashed line's position with the mouse pointer.

The amount (percent) of blurring at the edge of the Rotate Circle can be controlled by the value specified in the **BRadius(0-99)** input field. This is represented on the Rotate Circle as a small vertical line (marker) on the radius line. A value of **0** (marker at the edge of the Circle) produces no blurring, whereas a value of **99** (marker at the center of the circle) produces blurring across the entire radius of the Circle. The position of the blur radius marker, as well, cannot be modified with the mouse pointer.

The speed (and correspondingly, the quality) of the rotate operation can be controlled with the **Quality** button. Clicking on the button will toggle the type of quality between **Fast** (not the best quality, but faster to complete) and **High Quality** (high quality, but slower to complete).

To abort the display of the current image, select the **Abort** button. Only a portion of the entire image may be shown on screen, but you can still move the Rotate Circle almost anywhere on the image.

To accept the current Rotate Circle values and begin the operation, select the **Accept** button. A meter window will appear on the Rotate control screen, showing the progress of the operation. To abort the operation at any time, select the **Abort** button.

If you have finished modifying your image for this session and want to return to the MorphPlus screen, select the **Quit** button.

7.15 Scale

When 8 bit-plane gray or 24 bit-plane color data is present in memory, you may digitally reduce or enlarge the width or height (or both) of the image. This is useful in a number of ways. For example:

- The individual pixels in an NTSC screen are not square. In fact, they are higher than they are wide in low resolution, non-interlaced displays. However, the pixels produced by many image sources such as scanners and 3D modelling programs are perfectly square. This mismatch in aspect ratio can result in NTSC images which appear vertically stretched. By reducing the height of an image to approximately 86 percent of its original size, you can produce images with an accurate aspect ratio for NTSC screens.[1]
- Various artistic effects can be accomplished by reducing the width and height individually.
- You can create low resolution interlaced images easily by reducing the width alone by 50 percent and then enabling interlaced mode.
- You can create high resolution non-interlaced images (such as in the preparation of icons for the Workbench) by reducing the height alone by 50 percent.
- You can cause an image to fit into any desired standard screen size. You can preserve the image's aspect ratio by reducing both the width and height by the same amount. Or, you can reduce the width and height by different amounts.
- You can produce a special effect called "pixelization" or "posterization" by significantly reducing the image and then significantly enlarging it.

Selecting the Scale operator when raw data is available in memory will cause the scaling control panel (shown in Figure 7.16) to appear.

You can switch between reduction and enlargement by selecting the button marked **Switch to Enlarge**. At any one time, you can either reduce or enlarge, but not both. When in the enlargement mode, the button which will switch you to reduction will be marked **Switch to Reduce**.

Using this control panel, you can specify a reduction or enlargement in two ways:

1. You can specify the number of pixels to be the resulting width and/or height using the supplied integer input fields.
2. You can specify the percent of reduction (or enlargement) of the width and/or height using the slider or input field.

When using the keyboard to enter values, the **RETURN** key will toggle between the width and height percentage of change or the new width and height in

[1] Since the actual aspect ratio of a given screen will vary from monitor to monitor, the figure of reducing an image's height by 14 percent is *only approximate*. Also, PAL video is much closer to square than NTSC. Therefore, PAL screens may not require adjustment at all.

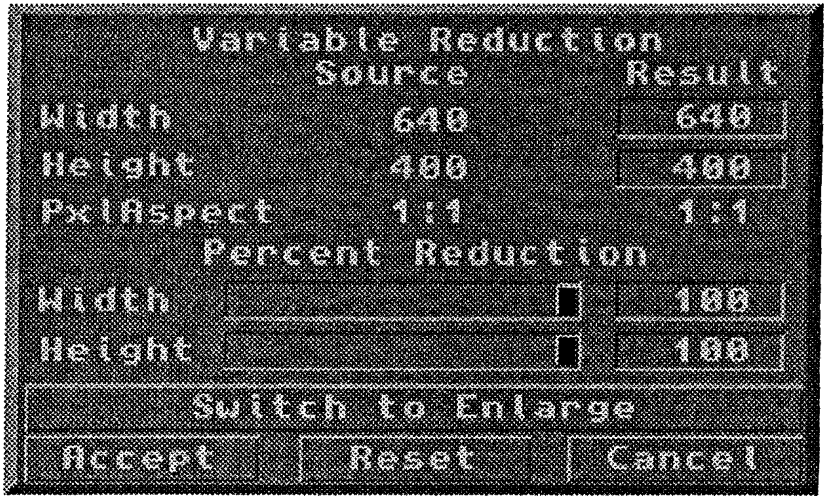


Figure 7.16: The Scale Operator's control panel.

pixels depending upon which pair of gadgets are active. `Alt-RETURN` (pressing the `RETURN` key while holding down either `Alt` key) will toggle between the pairs of input fields.

To reset the values to what they were when the control panel first appeared, you can select the **Reset** button. To ignore any scaling you might have specified, you can select the **Cancel** button.

Note that using the sliders, you can only specify reductions down to 25 percent of original size, or enlargements up to 400 percent of original size. However, by entering the scaling amounts manually, you can define reductions to less than 25 percent of original size, or enlargements to larger than 400 percent of original size.

To accept a set of reduction or enlargement values, simply select the **Accept** button or depress `Shift-RETURN` from the keyboard. *Unlike other operators, scaling operations are initiated immediately after acceptance.*

Quality digital scaling requires massive computation. The computation time of MorphPlus' scaling facility is proportional to the size of the resulting image. Also note that scaling permanently alters the raw data in memory. After an image has been scaled, you would have to reload from disk in order to get back the unscaled data.

7.15.1 Pixel Aspect

The IFF format supports the storing of a pixel aspect ratio with the file. When MorphPlus loads an IFF-ILBM file containing this pixel aspect ratio specification, it presents this information in the Scale control panel as shown in Figure 7.16. The pixel aspect defaults to 1:1 (pronounced "1 to 1") when MorphPlus loads a file which does not contain any pixel aspect information.

MorphPlus maintains the pixel aspect information as scaling operations take place. The aspect shown in the **Source** column in Figure 7.16 represents the current pixel aspect. The aspect shown in the **Result** column represents the pixel aspect which would result from the currently defined scaling operation if it were to be executed.

This information can be used to your advantage especially if you know that the device this picture is intended for display (or printing) has a specific pixel aspect. By adjusting the scaling controls so that the pixel aspect shown in the **Result** column matches the intended pixel aspect of the device, the aspect of objects within the image will be preserved.

For more information about pixel aspect, please see Section 2.7.

7.16 Sphere

With the Sphere operator, you can distort your rectangular shaped images into a spherized, or domed, shape. It is as if you were looking at your image through a glass dome. The rectangular border of the image starts to conform to a circular shape, and the middle portion appears to "buckle" upward.

You are not limited to producing sphered images with circular edges. You can also create spheres with edges that are less than circular. Specifically, the corners of a rectangular image can be incrementally "bent" toward the center of the image, with a final shape being a curve.

Select the Sphere operator from the MorphPlus operator list. Please refer to Section 2.4.1 for specific instructions.

After selecting the operator, a preview screen should appear with the Sphere operator control panel on it.

7.16.1 Sphere Control Panel

The Sphere control panel can be displayed by selecting the **Open...** menu item in the **Control** menu. To close it, either select the **Close** menu item (in

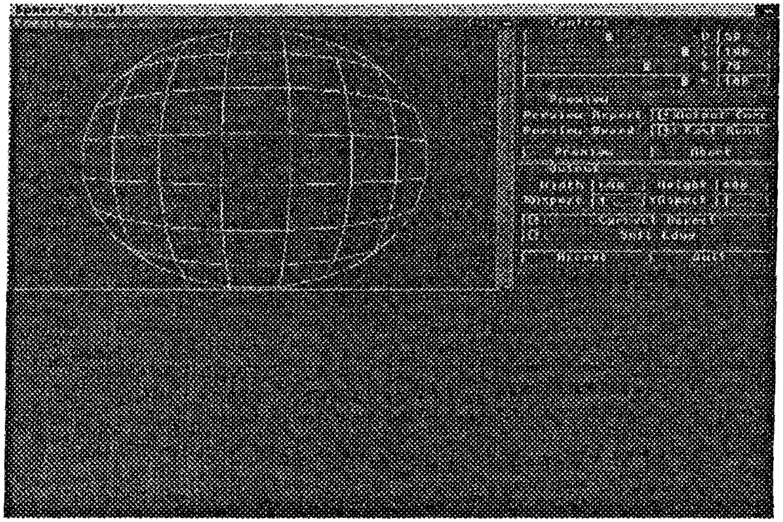


Figure 7.17: The Sphere Operator screen.

the same menu) or click on the window's close gadget.

The **D** (distortion) control defines the amount of sphering that will be done to the image. A value of 0 will not spherize the image at all, while a value of 100 will severely spherize the image, probably showing only one pixel's worth of image data (which is probably not what you want). The higher the number, the more the distortion.

The **C** ("circle"-ness) control defines the shape of the edge of the image. With a **C** value of 0, the image will be sphered but the edges will not follow a circular shape (as a half sphere would). With increasing values of **C**, the edges will begin to look more like a circle.

The **S** ("square"-ness) control defines how square the image should be. If this value is set to 0 and **C** set to 100, the edges may have curved edges, but it might still look like an oval. Higher values of this control will make the rectangular image more square, so that the result will look more circular than oval.

The **%** (% effect) control defines what percentage of the settings should actually be used to transform the image. For example, you could have **D**, **C**, and **S** all at a value of 100, but if your **%** value is only at 50, only 50% of the intended effect will be shown. Usually, you will not need to change this from the default value of 100.

7.16.2 Preview Control Panel

The Preview control panel can be displayed by selecting the **Open...** menu item in the **Preview** menu. To close it, either select the **Close** menu item (in the same menu) or click on the window's close gadget.

The pixel aspect of the preview rendering (shown in the **Preview Aspect** cycle gadget) can be specified by selecting **No Corr**, **1:1 Corr**, or **Output Corr**. **No Corr** will not correct the pixel aspect. **1:1 Corr** will display the image as if it were being displayed on an output device that has a 1:1 pixel aspect. **Output Corr** will display it using the currently defined Output pixel aspect (**XAspect** and **YAspect** in the Output control panel).

The quality of the preview rendering can be controlled by selecting either **Good Rend** or **Fast Rend** from the **Preview Speed** cycle gadget. **Good Rend** produces a better rendering but is slower, whereas **Fast Rend** is faster but doesn't render (in the Preview window only) as well. If you want to get a rough idea of the current sphere settings, use **Fast Rend**.

To preview the current settings, select the **Preview** button. The image that would be produced with these settings is shown in the Preview window. To abort the preview rendering at any time, select the **Abort** button.

To reset only the above values to the values that were shown at the time this operator was selected, select the **Reset** menu item in the **Preview** menu.

7.16.3 Preview Window

The Preview window displays the image as it would look if the current settings, defined in the Sphere, Preview, and Output control panels, were used.

The width and height of the resulting image (which may be larger than the area occupied by the original image) is represented by the inner border of the window.

The window can be shrunk or enlarged, but it will always preserve the relationships among the **Width**, **Height**, **XAspect**, and **YAspect** values. This will cause the window to adjust itself to the proper size, even though you resized it differently.

7.16.4 Output Control Panel

The Output control panel can be displayed by selecting the **Open...** menu item in the **Output** menu. To close it, either select the **Close** menu item (in the same menu) or click on the window's close gadget.

The width of the resulting image can be set by changing the value in the **Width** input field. The height is set in the **Height** input field. These two values let you modify the size of (or scale) the image, which most operators do not let you do. Changing either of these two values will automatically adjust the Preview window's size to properly represent the new size.

The rendered preview can be displayed using a user-defined pixel aspect by entering values in the **XAspect** and **YAspect** input fields. This can be very handy for when you want the rendering in the Preview window to match the pixel aspect of a given output device.

The pixel aspect, as defined in MorphPlus and modifiable with the Define_Pxl_Aspect operator, can either be ignored or corrected for by selecting either **Ignore Aspect** or **Correct Aspect**, respectively. **Ignore Aspect** will not use the MorphPlus-defined pixel aspect, but instead will assume a 1:1 aspect. **Correct Input** will take this value into account when rendering the preview.

The edges of the image may be anti-aliased by setting the **Edge Type** to **Soft Edge**. **Hard Edge** will not anti-alias the edges. **Soft Edge**'s main use is to create grayscale images for use as alpha channels in programs, such as Art Department Professional, that support them. These alpha channels can be created with **Soft Edge** to smooth out the aliased edges that a sphered image (generated by the Sphere operator) may produce. These alpha channels are great for compositing one image over another, without the "hard edged" look that can result.

To reset only the above values to the values that were shown at the time this operator was selected, select the **Reset** menu item in the **Output** menu.

If you are satisfied with the current settings and want to actually modify your image, select the **Accept** button or menu item (in the **Project** menu). To return to the MorphPlus screen without modifying the image, select **Quit** (either button or menu item).

7.16.5 Other Controls

Although you can selectively reset the attributes in any of the panels, to reset all attributes to their default or last saved values, use the controls in the **Settings** menu. **Reset to Default** resets all controls to their default, never-changed, values; **Last Saved** resets them to the values they had the last time the operator was used, and hence, the values at the time the control screen and panels were opened (for the current session).

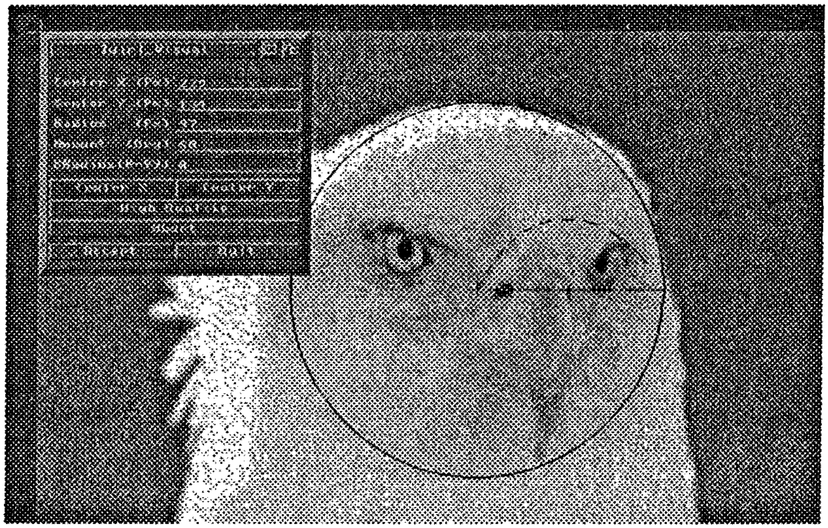


Figure 7.18: The Twirl Operator screen.

7.17 Twirl

With the Twirl operator, you can twirl circular areas of your image. The direction, center, amount, and quality of twirl, as well as the amount of blurring used at the edge of the circle, can be controlled.

Select the Twirl operator from the MorphPlus operator list. Please refer to Section 2.4.1 for specific instructions.

After selecting the operator, a preview screen should appear with the Twirl operator control panel on top of a grayscale representation of the currently loaded image.

The Twirl Circle visually describes the circular portion of your image that will be "twirled". As you can see in Figure 7.19, the Twirl Circle is comprised of a thin circle with a crosshair at its center, a horizontal radius line in the right half of the circle, a small vertical line on this radius line, and an dashed arc extending from the center of the circle out to the edge.

Although you can enter values into the control panel's input fields, the more intuitive way of specifying the location of this twirl area on the image is to move the center crosshair on the Twirl Circle. Moving this crosshair moves the Circle as a whole around the image. To do this, place the mouse pointer above the crosshair, hold down the left mouse button, move the mouse so that the crosshair moves to the new location, and release the hold of the mouse

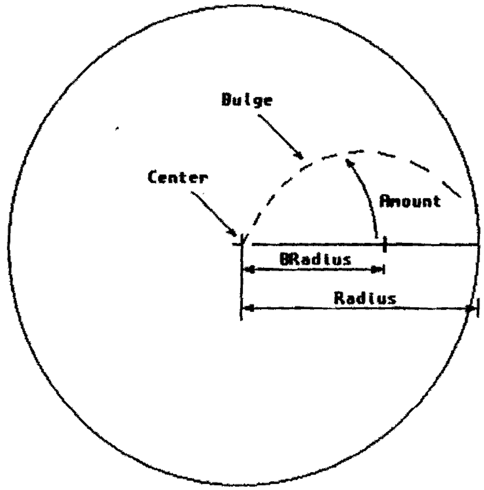


Figure 7.19: Components of the Twirl Circle.

button. Notice that when you move it around the image, the values in the **Center X (Px)** and **Center Y (Px)** input fields change as well. This gives you real-time feedback of your changes.

To center the horizontal position of the Twirl Circle, select the **Center X** button. To center the vertical position, select the **Center Y** button.

Now that you've selected the center of your twirl area, you will want to specify the radius of this Circle. If you click the left mouse button while the pointer is anywhere but on the crosshair, the radius of the circle will snap to the new location. The actual radius (in pixels) is shown in the panel's **Radius (Px)** input field.

To define how much twirling will be done, you will have to enter a value in the **Amount (Deg)** input field. The current amount of twirl is shown on the Twirl Circle as a "bulge" or arc from the radius line in the direction the twirl will take. This bulge also shows in what direction (clockwise or counter-clockwise) the area defined by the Twirl Circle will be twirled and by how much (in degrees). If a positive **Amount (Deg)** is specified, the arc will bulge upward and the twirl will be in a counter-clockwise direction. If a negative **Amount (Deg)** is specified, the arc will bulge downward, producing a twirl in the clockwise direction. Note that you cannot modify the bulge with the mouse pointer.

The amount (percent) of blurring at the edge of the Twirl Circle can be controlled by the value specified in the **BRadius(0-99)** input field. This is repre-

sented on the Twirl Circle as a small vertical line (marker) on the radius line. A value of **0** (marker at the edge of the Circle) produces no blurring, whereas a value of **99** (marker at the center of the circle) produces blurring across the entire radius of the Circle. The position of the blur radius marker, as well, cannot be modified with the mouse pointer.

The speed (and correspondingly, the quality) of the twirl operation can be controlled with the **Quality** button. Clicking on the button will toggle the type of quality between **Fast** (not the best quality, but faster to complete) and **High Quality** (high quality, but slower to complete).

To abort the display of the current image, select the **Abort** button. Only a portion of the entire image may be shown on screen, but you can still move the Twirl Circle almost anywhere on the image.

To accept the current Twirl Circle values and begin the operation, select the **Accept** button. The progress meter window will appear, which shows an increasing (from left to right) meter bar describing the completeness of the operation. To abort the operation at any time, select the **Abort** button.

If you have finished modifying your image for this session and want to return to the MorphPlus screen, select the **Quit** button.

7.18 Vertical_Flip

Many applications require the production of mirror images of the original data. Specifically, this is required by screen and heat-transfer printers. Selecting the Vertical_Flip operator will vertically flip the raw image data currently in memory.

Note that performing a horizontal flip followed by a vertical flip is the same as rotating the image 180 degrees. This can be used in conjunction with the orientation (**Port / Land**) button to rotate images through 0, 90, 180, and 270 degrees. Note, as well, that you can also do these same rotations with either the Rotate or Perspective operators.

7.19 Warp

The Warp operator lets you distort (warp) areas of a single image. To do what is more popularly known as morphing—transforming one image into another—you will need to use the separate program, Morph. This program uses the Warp operator to assist in the morphing.

Select the Warp operator from the MorphPlus operator list. Please refer to

Figure 7.20: The Warp Operator screen.

Section 2.4.1 for specific instructions.

The operator will open a separate control screen (shown in Figure 7.20) containing a grayscale representation of the current image occupying most of the screen. Unlike most of the other operators that have main control panels, most of the Warp controls are located in the menu bar. Hold down the right mouse button to display and scroll through the available menu items.

Some of the controls, however, are right on the screen itself. Two of these controls are the brightness and contrast of the grayscale image used in the visual user interface. The **Brt** lets you set the brightness of the grayscale representation. The actual image is not modified, only the on-screen display. The lower the slider value, the darker the image, and vice versa. Contrast can also be controlled with the **Cnt** slider. These two sliders work similarly to the sliders in the Balancing control panel (see Section 4.3.1).

7.19.1 Vectors

The best way to describe what this Warp operator will do is to think of the image as a rectangular sheet of rubber. This sheet has a certain elasticity to

it, with portions of it able to expand and contract. When you warp an area of the image, you are actually stretching or compressing this "rubber sheet".

The Warp operator allows you to specify the "movement" or warping of this sheet. To warp an area of the image, you would specify a vector which has a start point and end point. The start point describes the initial position of that point on the image before the warp. The end point describes the final position of that same point. This vector describes in what direction and how much the image should be warped.

Think of our sheet of rubber analogy again. If you were to poke a dull-tip pencil through the bottom and directed to the right side of the sheet, the area to the left of the pencil and at the pencil tip would be stretched proportional to the distance the pencil's tip stretched the sheet of rubber. This is the same effect that a vector will produce on your image, although the area around the vector might not be as much affected as in the case of an actual pencil.

To create a new vector definition, select the **New Vector** menu item. The newly created vector will follow the movement of the mouse pointer. Move the mouse pointer to the location on the image that you want to place the vector, then click the left mouse button to place it down. The start point is shown as a filled (solid) red square, with the end point as an unfilled (only the outline) red square. A red line connects the centers of both boxes.

You should also notice that both start and end points have outlines around them and the line connecting them is more than one screen pixel thick. This is how a selected vector looks like. When it is not selected, the outlines are not shown and the line is thinner. Selecting and deselecting vectors will be discussed later in this section.

A special case vector, called a "point", is one which has the same location for start and end points. The effect this point has is, if we can use our analogy of a rubber sheet, of a nail or staple through the sheet. If you want to restrict the movement of a point in the image, use a point. This is created with a **New Point** menu item. Note that you cannot change a point back into a vector, although you can do the opposite. Making a vector into a point does not force the vector to become unchangeable, as with a created point. To help you distinguish a vector's start and end points from a point, the point is drawn as a solid circle, as opposed to the square shape of a vector's points. For the rest of this manual, the term "vector" will be used to refer to both vectors and points unless otherwise indicated.

To flip (exchange) the start and end points of the currently selected vector(s), select the **Flip Selected** menu item. You might need to do this if you placed your vector in the direction opposite of what you intended. This operation has no effect on points.

To delete the currently selected vectors, select the **Cut** menu item (from the

Vectors menu). This command places the descriptions of the deleted vectors into the standard Amiga Clipboard.

NOTE: If you use the `Del` *(delete) key to "cut" vectors, the deleted vectors will NOT be copied to the Clipboard.*

To copy (but not delete) the currently selected vectors and edges, select the **Copy** menu item. Like the **Cut** command, **Copy** will place the copied vector definitions to the Clipboard.

To paste the current vector definitions from the Clipboard onto the screen, select the **Paste** menu item. If the Clipboard contains data in the appropriate format, it will create vectors for them. The objects that will be "pasted" will show up attached to the mouse pointer. You can then place them down on the image in the same way as described earlier.

Selecting and deselecting of vectors can be done several ways. Clicking on either the start or end point of a vector only selects that point, but the vector itself is still considered selected. Clicking anywhere on the line connecting the points selects that vector. If the vector was previously selected, clicking on it causes it to be deselected. Points, having both start and end points the same, will either be, as a whole object, selected or unselected. To select all of the currently defined vectors (that are visible), select the **Select All Visible** menu item. You can also select only those vectors which are connected (explained later in this section) to other vectors with the **Select All Connected** menu item. The **DeSelect All** menu item deselect all vectors, whether selected or not.

You also have keyboard shortcuts for creating and selecting vectors and points. To create multiple new vectors without having to select a menu item or press a button for every one, hold down the `n` key and start clicking (using the left mouse button) on the areas of the image you want to place new vectors.. To stop this mode, release the `n` key. To create multiple points, use the `m` key instead.

To select multiple vectors without having to click on each vector and point, hold down the `s` key and *slowly* drag the mouse pointer above the vectors and points you want selected. If you cross the vector (and neither of its points), both points will be selected; once you pass over a point, it will become selected. If you cross either point of a vector, only that point will be selected. To stop multiple-select mode, release the `s` key. As was stated before, all selected vectors have their points outlined and their lines drawn thicker. Points, as is the case with a vector's start and end points, will be outlined when selected.

You can selectively display all or parts of vectors in several ways. Both start and end points are displayed by selecting **As Vectors** from the **Vectors Display** menu item in the **Settings** menu; only start points are displayed by se-

lecting **As Start Points**; and only end points show up when **As End Points** is selected.

The size of a vector's start and end points and of points themselves can be resized by selecting either the **Large** or **Small** menu subitem (from the **End Points** menu item under the **Settings** menu). **Large** points might be useful to better identify the vectors on the grayscale "background", whereas **Small** points might come in handy when many vectors and points are in a small area.

You can also hide selected vectors, for those instances when you are using many vectors in a small area and want the screen to be less cluttered, by selecting the **Hide Selected** menu subitem (in the **Vector** menu's **Visibility** menu item). To show all defined vectors, select **Show All**.

These hidden vectors can be displayed in two ways: ghosted or invisible. By selecting **Invisible** from the **Hidden Vectors** menu item, all hidden vectors will be totally hidden from view. However, if you select **Ghosted**, the vectors will be drawn as a lightly-dashed line. Although the vector is still somewhat visible, it cannot be selected.

Each vector is given a unique vector number. To display these numbers, select **Display Numbers**. The number will appear next to the end point (or the point itself for points) and is used to distinguish one vector from another.

NOTE: The vector number is mainly for your convenience. The Warp operator does not use this number to determine the order in which the warping will be done. All of the vectors and points will be processed at the same time.

To move either start or end point of a vector, first make sure the point is not currently selected. Move the mouse pointer above the point, hold down the left mouse button, and drag the mouse around the screen. You will notice the point follow the location of the pointer. To move the entire vector—both start and end points at the same time—drag the line that connects the two points. This will select the both points of the vectors and move them in tandem.

Actually, any vectors (or points) that are currently selected will move in tandem with the mouse pointer. This allows you to select a lot of vectors (by `Shift`-selecting them or using the `s` key) and collectively move them around the screen.

If you find that you want to abort the movement of a collection of vectors in progress, while still holding down the left mouse button press the right mouse button.

The screen can become cluttered if you have a lot of vectors in a small area. It can be difficult to spot the currently selected vector. The **Flash Selected** menu subitem (in the **Vector** menu's **Visibility** menu item) can be used to

flash all currently selected vectors four times in succession. Note that the fewer the number of selected vectors (and edges between them) the faster they will flash, and vice versa.

Not only can the currently selectable vectors be flashed, but also the hidden ones as well. Just as with **Flash Selected**, you can use the **Flash Hidden** menu subitem to flash the hidden vectors.

Another way of determining the currently selected (or last selected, if none or many are selected) is to look at the contents of the input fields at the bottom of the screen. The **Vector** input field will display the selected vector's number, with the vector's location shown in the **Start X, Y, End X, Y, Delta X,** and **Y** fields. You can modify the vector by changing these values instead of using the mouse to drag the start and end points. The group of which this vector is a member is displayed in the **Group** text field; newly created vectors are initialized to the "Default" group, unless otherwise modified. The total number of vectors currently defined is shown in the text field below the **Vector** input field.

If you want to place vectors very close to one another but the magnification of the image does not allow for it, use the subitems available in the **Zoom** menu item of the **Settings** menu. After selecting **In**, the mouse pointer will change to a magnifying glass with a plus sign in it. Move this pointer above the area you want to zoom into and click the left mouse button. The area will be magnified by a factor of 2 to 1. If you hold down the `Shift` key while selecting **Zoom**, you will get a 3 to 1 zoom. If you hold down the `Alt` key (either one), you will get 4 to 1. Holding down the `Ctrl` (Control) key while selecting **Zoom** will zoom in all the way (as much as possible).

To zoom back out one level, select the **Out** subitem. The `Shift`, `Alt`, and `Ctrl` keys use the same level of magnification zooming out as they did zooming in. To zoom back out to the original (no) magnification, select **Full Size**. Note that using the `Ctrl` key with the **Out** subitem is equivalent to using the **Full Size** subitem. If you selected **In** but want to cancel the operation, press the right mouse button and the zoom pointer will revert to the normal pointer.

While in zoom mode, you can move around the image using the cursor (arrow) keys. The `UP`, `DOWN`, `LEFT`, and `RIGHT` keys will move the area being magnified in the up, down, left, or right direction, respectively. Although this will move the area in small increments, to move in screen-size increments hold down the `Shift` key while pressing a cursor key. Using the `Alt` key moves in even larger increments or the `Ctrl` key to move the maximum distance in a given direction.

A very useful feature of the Warp (and of the Morph program as well) is that the drawing operations are asynchronous—you do not have to wait for the screen to finish redrawing before selecting another operation. For instance, let's say you want to zoom into an area of the image, but you selected the wrong area. The screen will start to display the selected area, but before it

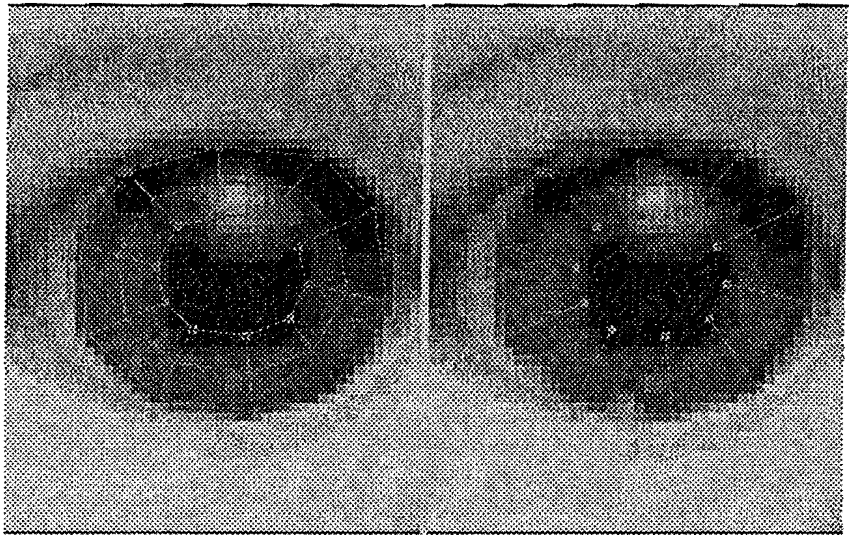


Figure 7.21: Vectors with (left image) and without (right image) edges.

finishes, you can repeatedly press any of the cursor keys to move to the area you intended. This allows you to spend more time working on your warp, and less time struggling with the user interface.

A handy feature is the ability to load, merge, and save the currently defined vector definitions. The **Open...** menu item allows you to load a previously saved set of vectors; this menu item will delete all currently defined vectors before loading in the file's vectors. If you want to preserve the current vectors and merge in another set, use the **Include...** menu item. To save the current set of vectors, select the **Save All As...** menu item which allows you to save the vectors under the same or different name. In all operations, a file requester will appear for you to select the file name of a vector definitions file.

7.19.2 Edges

As you can see in Figure 7.21, two vectors spaced far enough apart from one another with another vector "pulling the sheet" in the opposite direction may produce unwanted results. One way this problem can be solved is by placing more vectors inbetween these two to guarantee that the line connecting the two start points will move just as the points themselves do. An easier solution is to use "edges". Edges, as shown in Figure 7.21, are dashed lines connecting the start, middle, and end points of any two vectors. This will produce the effect of an infinite number of intermediate vectors, without having to define

them (and clutter the screen). Once an edge is created between two objects (vector to vector, vector to point, or point to point), the objects are considered to be "connected".

To create an edge between two vectors, first select one of the vectors. While holding down the , (comma) key, select the vector which will make up the other end of the edge. If the two vectors already have an edge between them, pressing , will remove the edge.

Adding an edge between two points has the effect of a long series of "nails" or needlepoint stitches through the sheet of rubber. You are guaranteeing that these points will not be affected by vectors around them. The edges are drawn as dashed lines as a visual cue that they are not to be confused with vectors (drawn as solid lines). In fact, using a dashed line will help you think of them as stitches of thread through fabric, where the fabric in this case is a sheet of rubber.

NOTE: Holding down the **,** *key while clicking on a vector will define an edge between the current vector and the* last selected vector, *whether the previous vector is selected or not.*

A time-saving feature is the ability to create new, connected vectors. This is done by holding down both the **n** and **,** keys and successively clicking on the left mouse button. A new vector will be created, but it will also have an edge to the previously created (using this method) vector. To stop using this feature, simply release both keys. You can also create connected points using a similar procedure, but using the **m** key instead of the **n** key.

You can select just the connected vectors by selecting the **Select All Connected** menu item.

The area underneath edges acts in different ways depending upon what is on each end of the edge. An edge between points is opaque, whereas an edge between vectors is not. Think of opaque as the ability of part of the image on one side of the edge to affect the image on the other side—opaque means that one side doesn't affect the other and non-opaque (or clear) means that it does.

If an edge connects two hidden vectors, that edge is also hidden. How these hidden edges are drawn on the screen is controlled by the option that is selected in the **Hidden Edges** menu item (under the **Settings** menu). If **Visible** is selected, then these edges will be visible, whereas if **Invisible** is selected, then they will actually be hidden.

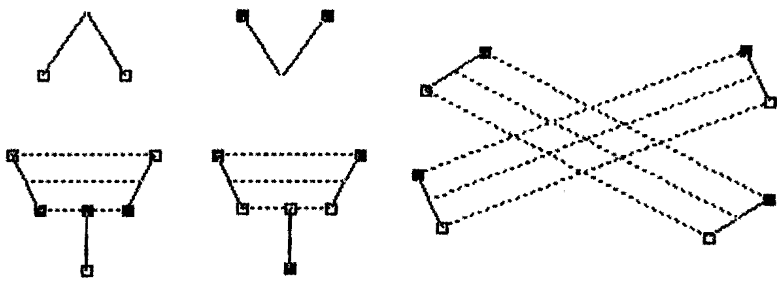


Figure 7.22: Vector and edge placements that you should avoid creating.

7.19.3 Vector and Edge Restrictions

Although you have complete freedom to place vectors and edges anywhere on your image, there are a number of restrictions which should be noted. Figure 7.22 illustrates these cases.

- The start point (of a vector) should not be on top (same location) of another start point. (Upper left pair of vectors. Note that two start points drawn at the same location will cause no point to be displayed—they cancel each other out).
- The end point (of a vector) should not be on top (same location) of another end point. (Upper middle pair of vectors.)
- The start point should not be on top of an edge between other start points. (Lower left set of vectors.)
- The end point should not be on top of an edge between other end points. (Lower middle set of vectors.)
- Edges between start points should not cross edges between other start points. (Rightmost set of vectors.)
- Edges between end points should not cross edges between other end points. (Rightmost set of vectors.)

Every effort is made by the Warp operator to filter out these special cases, but knowing the limitations beforehand will make it easier for you to create proper looking images.

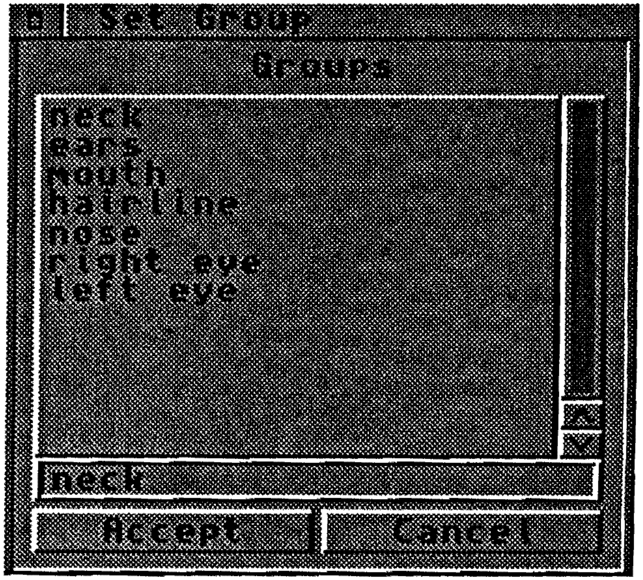


Figure 7.23: The Set Group control panel.

7.19.4 Groups

Most of the time when you are trying to warp an image, you are distorting individual features, such as making an eyeball bulge or nose enlarge. You probably have specified these vectors as the "eyeball" vectors or the "nose" vectors. When you work on these vectors, you are either moving a feature's vectors as a group. The Warp operator allows you to characterize vectors in terms of groups of vectors, allowing you to do many operations on a group-by-group basis, rather than vector-by-vector.

Groups, in Warp, can be given descriptive names. All the vectors of a particular group share the same attributes, such as speed, transparency, and depth; all of these attributes will be described later in this section.

Initially, all vectors belong to the "Default" group. To place a set of vectors into a different group, select all of the desired vectors then select the **Add Selected To Group...** menu item (in the **Vectors** menu). The Set Group control panel will appear (shown in Figure 7.23), from where you can select a name for the new group. Also, if you want to place these selected vectors into a new group, instead of typing a new group name, just select the name from the list of currently defined groups.

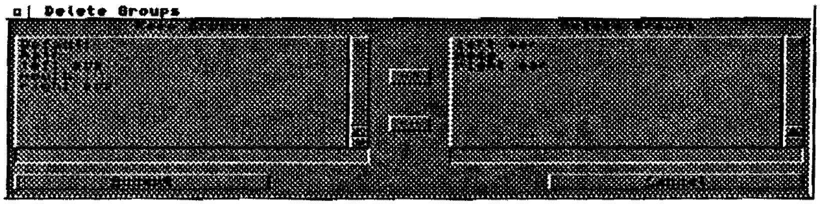


Figure 7.24: The Delete Groups control panel.

Selecting all the vectors of a particular group is made simpler with the use of the **Select Members** menu item. Let's say that you want to select all vectors belonging to the "left eye" group. Instead of individually selecting these vectors, select one of the vectors, then select the **Select Members** menu item. All of the "left eye" vectors will then be selected.

Once a group or set of groups have been selected, you can use the previously described **Cut**, **Copy**, **Paste**, and **Erase** menu items (in the **Vectors** menu) to do Clipboard-type operations.

An alternate way of erasing a group of vectors is to use the **Erase...** menu item (in the **Groups** menu). A control panel (as shown in Figure 7.24) will appear, from which you can move groups from one list to another. Double-clicking on a group entry in one list automatically switches it to the opposite list in the panel. Once you press **Accept** from this panel, those groups, whether selected or not, will be removed.

Sometimes, you might use a particular set of groups all the time. These groups can be saved to a file as a group definition file with the **Save All As...** menu item. To merge a predefined group definition file with the current set of groups, select the **Include...** menu item and select the file from the file requester that will appear.

Sometimes you might want to unclutter the screen, especially if a lot of vectors are being displayed. Although you can selectively hide vectors, you can also hide groups of vectors. In the **Visibility** menu item is a subitem called **Choose...**. This menu item, when selected, will bring up the Group Visibility control panel (shown in Figure 7.25). Just as with the Delete Groups control panel, you move group names between two lists; in this case, you are moving the names between the **Visible Groups** and **Hidden Groups** lists. Once you press the **Accept** button, those groups that are in the **Visible Groups** list will be visible, while those that are in the other list (**Hidden Groups**) will be hidden from view. If at any time you want to display all of the grouped vectors, select the **All On** subitem. Conversely, **All Off** will hide all grouped

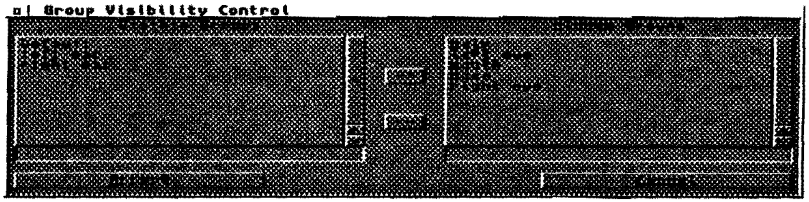


Figure 7.25: The Group Visibility control panel.

vectors.

Group Depth

One of the attributes that members (vectors) of a group share is their depth. Think of each point on your image as having a depth (z axis location). The higher the z value, the "closer" the point is to you, whereas the lower the z, the farther away it is. The actual image (and all vectors belonging to the "Default" group) can be thought of as being at $z = 0$. Note that this depth indication does not specify one vector being processed before another—all vectors and points are processed at the same time. It means that vectors at a higher z depth will obscure those at lower z depths, assuming that the higher ones overlaps the lower ones.

Take a look at Figure 7.26 for an example. The leftmost picture is the unwarped image. We intend for the mouth to be warped toward the top of the picture, while the nose to move downward, covering the area previously occupied by the mouth. The middle image has the "mouth" vectors' z depth set at 0 (the default value), while the "nose" vectors are at a higher depth. Notice how the nose properly obscures the mouth. The rightmost image, however, has all vectors at the default depth. Notice how the mouth "pokes through" the skin—not the result we intended. This is caused because the end points of the mouth vectors are not covered by the nose vectors. In fact, the mouth vectors are defined to end up above (larger y axis value) the nose vectors' start points. When the nose vectors are above the mouth, the mouth and lips are covered, but not when all the vectors are at the same depth.

Instead of making you remember numeric, and sometimes cryptic, values for z depth, the Warp operator lets you arrange the depth of groups relative to other groups. This is done with the Group Depth Order control panel (shown in Figure 7.27), which can be displayed by selecting the **Depth Control...** menu item.

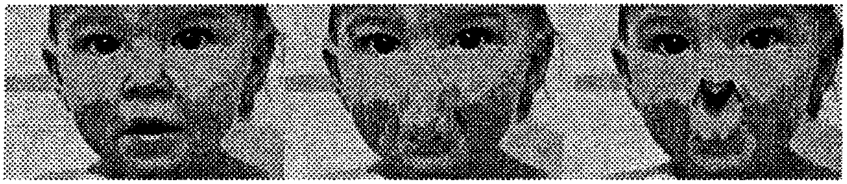


Figure 7.26: An example of how depth arrangement of groups can affect the final image.

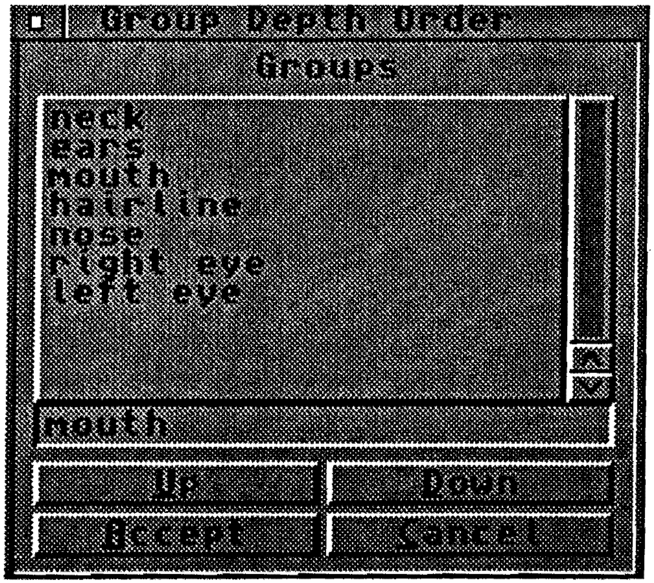


Figure 7.27: The Group Depth Order control panel.

Notice that it looks similar to the Set Group control panel—using a list of groups. Unlike this other panel, the Group Depth Order panel does not let you modify the group names (notice that the area below the list of groups is a text field and not an input field). Also, **Up** and **Down** buttons have been added. These let you arrange the currently selected group (shown in the text field) relative to other groups. The topmost item in the list corresponds to the highest z depth group, while the bottommost item corresponds to the lowest z depth. The "Default" group specification lets you arrange other groups with respect to the Default group.

Group Motion Control

All vectors in a group (including those that belong to the "Default" group) share the same amount of movement, or motion. Group motion describes how fast or slow the start points in a particular group's vectors will move toward their end points over time. Time, in this case, can be specified in frames or percent completion of the produced warp.

To specify this motion, select the **Options...** menu item (from the **Groups** menu). The Group Motion And Transparency Options control panel (as shown in Figure 7.28) will appear. This panel lets you specify the motion curve for each group, as well as the transparency curve. We'll discuss the Motion controls in the next few paragraphs and leave the Transparency controls for later.

Both motion and transparency curves are editable in the same panel, although only one can be modified at a given time. The current curve that is being edited is specified by which of the two radio buttons in the lower part of the panel (**Edit Motion** or **Edit Transparency**) is selected (depressed). To help you differentiate between the curves, the color of a curve corresponds with the color of the text that describes it.

Each group has its own pair of curves for motion and transparency. The name of the group that is currently being modified is displayed in the text field just below the **Group** list. To display and edit another group, simply select the group from this list.

Grid Usage

As shown in Figure 7.28, a majority of the panel is occupied by a grid. The y (vertical) axis describes the proportional distance from start to end point of each vector in the current group and the x (horizontal) axis describes the time component, where the leftmost part of the grid represents the beginning of time (start of the warp) and the rightmost part represents the end of time (completion).

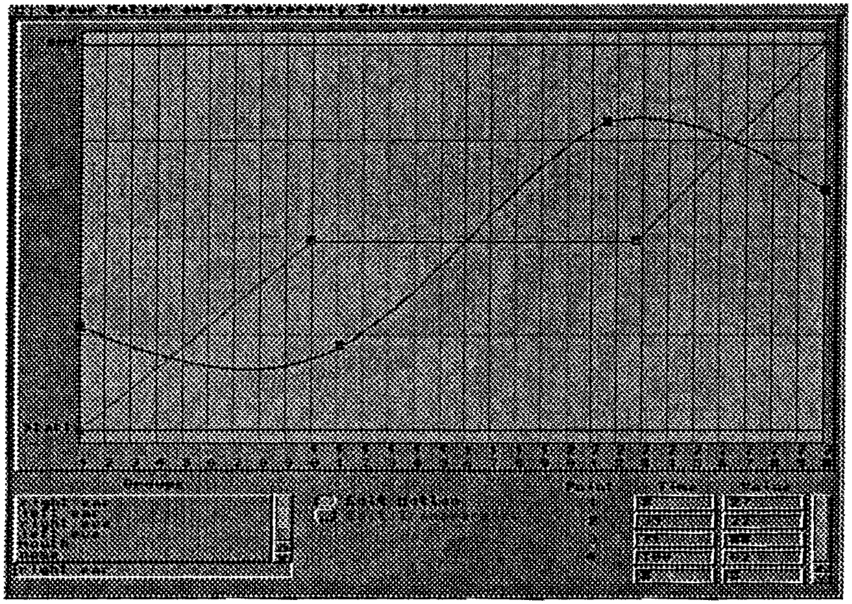


Figure 7.28: The Group Motion and Transparency Options control panel.

Each vertical line represents units of time, where time can be the total number of frames in the warp (**Display Time As** set to **Frames**) or percent completion of the warp (**Percentages**). If set to **Frames**, each vertical line will represent one frame in the resulting warp. If the number of frames is a large number, then each vertical line may not represent one frame, but several frames. This is done so that the vertical lines can be spaced far enough apart from one another for the frame indicators to still be readable. If the cycle gadget is set to **Percentages**, then there will be vertical lines for 0% to 100% and every 10% interval inbetween. In both cases, the lines are number appropriately.

For group motion, the horizontal lines (**start**, **end**, and the lines inbetween) correspond to the start and end points of each vector in a group. At any given time, the motion curve describes how far the start points will travel (as a percentage of total distance) to their end points. For example, let's say that at time = 25% done the curve is half way between **start** and **end** (a 50 percent warp). This means that at that moment in the progression of the warp, this group's vectors' start points (and surrounding areas) will have moved half the distance to their defined end points.

Curve Definition

The curve itself is defined by a set of points (shown on screen as small, solid squares) through which the curve will pass. Initially, the curve won't look like a curve at all, but a diagonal line starting at the source image side of the grid and ending at the destination image side. If you were to use this type of curve for group motion, then all the vectors in the current group would begin, at time 0, at the start points and end, at the last frame, at their corresponding destination points. Now let's assume that the line started at the **end** level and ending at the **start** level. This would make the vectors in this group move in the opposite direction (end to start points).

The curve has at least two points describing the start and ending points in the curve. To add intermediate points, select the **Add Point** menu item (from the **Edit** menu) and click the mouse pointer at the location on the grid where you would like to place a new point. The mouse pointer will change to a crosshair, signifying that you are in "add point" mode. The curve will immediately update itself, passing through the new point. A maximum of 30 control points can be defined for a curve. Removing a point is just as easy. Select the point that you would like to delete (the selected point has an outline similar to the outline for selected vector points), then select the **Remove Point** menu item. Note that you cannot remove the start or end points. Any of the points can be repositioned by simply grabbing them with the mouse pointer and dragging them to a new location. Although the intermediate points can be moved on either axis, the start and end points can only be moved along the y axis (up and down).

Notice how each of these bends in the curve are sharp, that is, the curve does not pass through the points smoothly. This sharpness might cause the motion to "ping pong". To produce smoother motion, you need to change the **Interpolation Method** indicator from **Linear** which produces sharp bends to **Spline** which produces smoother bends; look in the **Other** menu for the **Interpolation Method**'s menu item's subitems. For reference purposes only, the spline is a natural cubic.

If the method is set to **Linear**, then the curve between any two points will be a straight (linear) line. What this means is that between these two points the the motion or transparency (whichever is currently being defined) will be consistently increasing, decreasing, or staying the same. This is analagous to driving a car at the same speed between two locations. If set to **Spline**, then the curve will pass through the points without any sharp changes of direction at these points. The curve will be reshaped so that it will produce smoother changes in motion or transparency across the whole curve. This is analogous to driving a car from town to town without having to make any quick turns of the wheel.

To better understand how the **Spline** setting affects the motion (or transparency) curve, think of each point as a restricting peg. The curve will pass through each point and is required to be smooth through it. Sometimes, the

compositing curve will have to "overshoot" (extend above) the actual level (of the next control point) just to be able to pass through it smoothly. It can also "undershoot" (extend below) the level for the same reason. Each point should be able to affect the curve through other points just by its relative position to these other points. The best way of getting a feel for a spline is to add some points and move them around.

Although you can move the curve points around the grid, for more precise control you'll need a way of manually entering these points' coordinates. The lower portion of the control panel contains a list of 5 pairs (**Time** and **Value**) of input fields. You can alternatively enter the motion level values in these fields. If you have more than 5 points defined, you can scroll through the point definitions by dragging the scroll bar or clicking on either of the arrow buttons. The point is described by the number labelled to the left of each row. Note that the **Time** value of the first and last points in the curve cannot be modified.

You can copy the shape of the curve for use as another group's curve by using the **Copy** menu item. To retrieve a previously copied curve, simply select the **Paste** menu item. Whatever curve you're currently defining will be replaced (pasted).

You might notice some space above the **start** and below the **end** lines. This space is used so that the motion curve can extend past either line. When the curve extends below the **start** line (referred to as "undershoot"), then the vectors in the current group will effectively change direction. For example, if a group of vectors will enlarge an eye, specifying an undershoot in the curve will, for the same group of vectors, cause the eye to shrink at those sub-**start** points. Conversely, if the curve extends beyond the **end** line ("overshoot"), then the end points of the group's vectors will actually extend beyond the defined end location (in the same direction of movement).

A typical use for this undershoot/overshoot capability is to generate a series of frames where, taking our eye example from the previous paragraph, the eye starts off at its normal size, shrinks just a little ("undershoot"), gets warped to the defined end locations (between **start** and **end** lines), "bulges" past this location ("overshoot"), and finally comes to rest at the end location. The curve that describes this motion is shown in Figure 7.29.

If you find that when you are specifying your motion curve you don't have enough room to create a desired overshoot or undershoot section, you can move the **start** and **end** lines closer together. To do this, drag either line closer to the other. The grid and curve will be redrawn to the new proportions. This feature should give you enough flexibility in specifying most motion curves. Note that the areas outside of the **start** and **end** grid are measured in the same "units" as the area between them, although no additional horizontal lines are given.

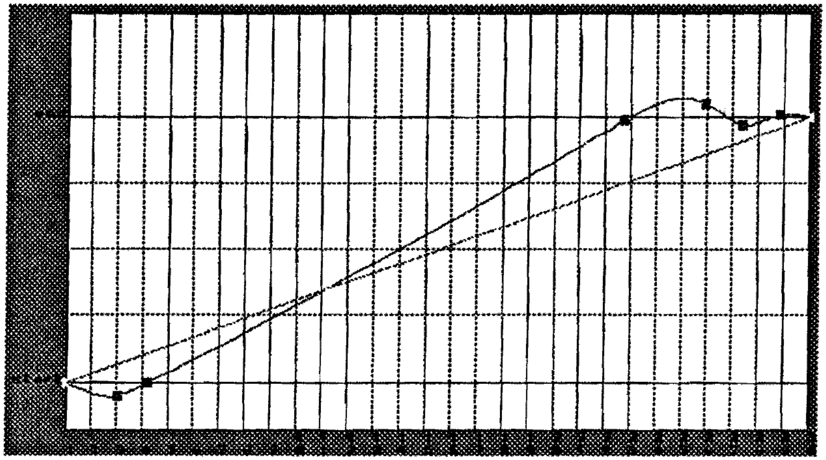


Figure 7.29: The motion curve for the bulging eye example.

In addition to being able to move these lines boundary lines closer together, they can also switch sides in the grid—**start** can be above **end**. The curve and control points will be redrawn in their new orientation, although the values in the list of input fields (described later in this chapter) will stay the same. This ability is given so that you can choose the curve representation that is easier to work with.

After some time with the Warp operator you might find certain curve work better than others. The **Open...** and **Save As...** menu items (from the **Project** menu) allow you to store and retrieve frequently used or interesting curve shapes. Note that **Save As...** will only save the current curve (motion or transparency) definition for the current group.

Once you are satisfied with the changes to the curve, select the **Close** menu item.

Group Transparency Control

The other curve that can be modified is the transparency curve (select **Edit Transparency**). All of the basic controls are the same as with the Motion curve, but only the meaning of the curve is different.

The transparency of a group of vectors describes how much of the effect of those vectors will be apparent at any point in time. How much a particular point is warped is independent of how much of it will show in the resulting image. Unless you are generating a transparency map (see Section 7.19.6),

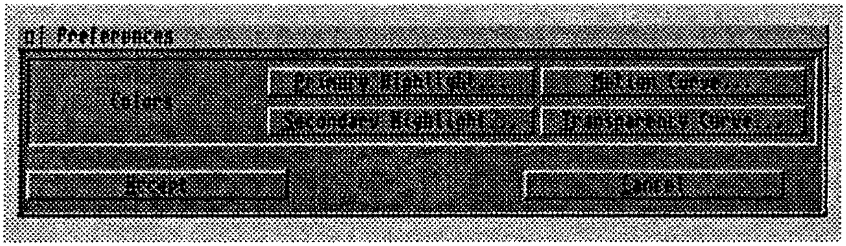


Figure 7.30: The Warp Preferences control panel.

then this curve will not mean much.

7.19.5 Preferences

The ability to customize your working environment can make your current job at hand go faster and smoother. The Warp operator allows you to set various parts of its environment with the Preferences control panel. Select the **Preferences...** menu item to display this panel (shown in Figure 7.30).

The color used for the vectors and edges can be chosen with the **Primary Highlight...** and **Secondary Highlight...** buttons. These two buttons let you specify your color options (two of them), although only one color will be used at any given time. Press either button to adjust the color for that particular attribute (primary or secondary color). You can toggle between the two highlight colors by repeatedly pressing the ` (back tick) key. The reason why more than one color can be selected is that, on some grayscale renderings of your images, the vectors do not stand out as well as other colors.

The color of the Group Motion and Group Transparency curves can similarly be specified through the use of the **Motion Curve...** and **Transparency Curve...** buttons.

Press the **Accept** button to retain and use the current Preferences settings, or press **Cancel** to preserve the previous settings.

7.19.6 Project Options

The Project Options control panel (displayed by selecting the **Options...** menu item) is where you control which frame of the warp you want to generate, as well as how the edges of the image will be handled. Figure 7.31 shows a picture of this control panel.

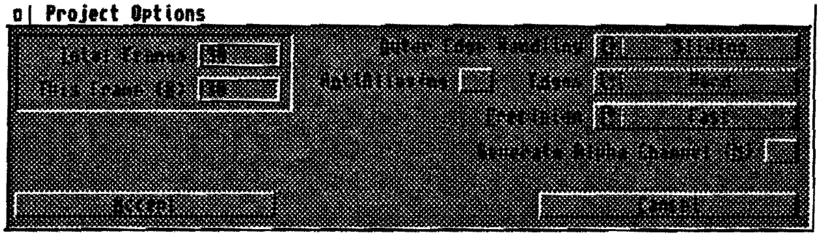


Figure 7.31: The Project Options control panel.

Edge Control

The outer edges of the image (its borders) and of a set of connected vectors can affect how the warp or morph will be done. By setting the **Outer Edge Handling** cycle gadget to a particular value, you can create different results using the same sets of vectors.

Fixed edges work as if the border of the image were nailed down. This allows you to place vectors close to the borders without having to worry about the image losing its rectangular shape.

Sliding is similar to **Fixed**, but it adds the ability for the parts of the image at the borders to slide along the border. A common use for this is when you are trying to warp the shoulders of a person inward (toward the center of the body). With a **Fixed** border, the "nails" at the bottom part of the image would cause the shoulder to stretch. With **Sliding**, the bottom border is allowed to slide toward the center of the body. For most cases, **Sliding** is the best setting to use since it provides this added feature while still allowing you to do everything that **Fixed** does. Also note that any or all of the borders can slide, and not just one of them.

Convex Hull allows you to remove (crop) areas of the image which do not lie within the convex hull of all the *connected* vectors; see Figure 7.32 for an illustration of the convex hull of a set of vectors. Think of convex hull as a rubber band around the entire set of vectors. This option is handy for when you want to "black out" the areas of the image which do not participate in the warp. Note that the image is cropped along the convex hull *before* the area is warped.

The **Cookie Cut** option is similar to **Convex Hull**, except that the connected vectors are cut along the vectors themselves and not just the convex hull of all connected vectors. For example, say that you have one set of connected vectors

Figure 7.32: The convex hull of a set of vectors. The right-side image was produced with Convex Hull edge type.

that affects the left eye, one that affect the right eye, one for the nose, and one for the mouth. With **Convex Hull**, the area that will be cut out would include the face features as well as the area between them. With **Cookie Cut**, only the features will be cut out, leaving the areas outside of the features set to black. See Figure 7.33 for a comparison between Convex Hull and Cookie Cut.

If some of your vectors are folding over others (see the discussion on Group Depth in Section 7.19.4), then you can specify (through the **Edges** cycle gad-

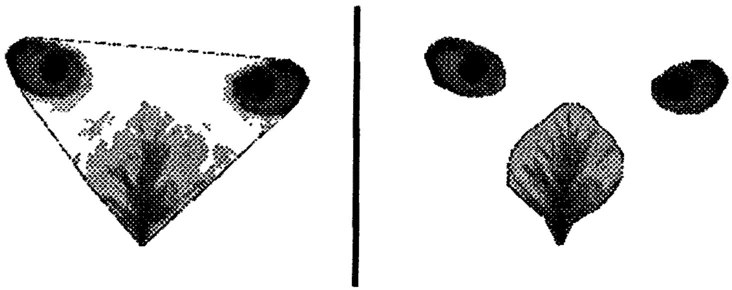


Figure 7.33: Convex Hull (left image) versus Cookie Cut (right image).

get setting) how smooth the folded areas will look as they "pass" over the underlying areas. With **Hard** edges, no smoothing will be done. With **Soft** edges, the edges will be anti-aliased.

The **AntiAliasing** check box, when checked, allows you to soften edges of **Convex Hull** or **Cookie Cut** edge types.

You may be asking yourself, "When should **Soft Edges** and **AntiAliasing** be used?" A good example for when to use one or both is when you want to create a "cut out" (with **Convex Hull** or **Cookie Cut**) and an alpha channel with the same set of vectors. To create the cut out, enable **AntiAliasing** but set **Edges** to **Hard**. This will produce a rough cut out. To create the alpha channel for the same image, you would process the image with **Edges** set to **Soft**.

Output Control

The quality or precision of the warping can be controlled with the **Precision** cycle gadget setting. **Fast** will perform the warp as quick as possible, use less memory, but sacrifice some quality. Conversely, **High Quality** produces a higher quality, but will use more memory and may take longer to complete.

With the Warp operator, you are given one image to modify. Unlike the Morph program where you can generate more than one frame, Warp only allows you to "generate" one, although the process of doing so is similar.

The **Total Frames** input field works as in the Morph—it describes the number of frames that *can* be generated. Used in the Warp operator, this field determines the total "time" needed to warp this image.

The **This Frame (B)** input field actually determines which frame from the total number of frames should actually be generated. The 'B' in parentheses is used as a keyboard equivalent (the same key that is pressed in the Morph Project Options control panel); think of it as the "beginning" frame (actually, the only frame) to generate.

An added feature of the Warp operator is its ability to create a image mask using the same vectors. If the **Generate Alpha Channel (M)** check box is checked and the operation is performed, the rendered image data will contain a 256 shade alpha channel of the image itself. The group transparency curves will define the generated grayscale image. This feature is best used with the Convex Hull or Cookie Cut modes, where you are enclosing an area of the image.

Once you are satisfied with the options, press the **Accept** button. **Cancel** will exit the control panel and restore the previous values.

To actually begin the warp, select the **Accept** menu item (from the **Project** menu). Selecting **Quit** exits the operator without performing the warp.

Chapter 8

ARexx Interface

By using the powerful scripting language, ARexx, MorphPlus becomes an enormously flexible automated image processing system. Nearly every aspect of MorphPlus can be controlled from ARexx using simple English-like commands. This chapter describes the usage of MorphPlus from ARexx. For specific ARexx command specifications, please refer to the various sections in Chapter 12.

NOTE: You must have ARexx in order to use the facilities described in this chapter and in Chapter 12. Specifically, if MorphPlus cannot locate the `rexxsyslib.library` *file (in the* `LIBS:` *directory), then it will not respond to any ARexx style requests.*

8.1 Addressing MorphPlus

In order to access MorphPlus or any of its Loaders, Saver, or Operators from ARexx, you must tell ARexx how to make contact with MorphPlus. Specifically, you must tell ARexx the name of the message port which MorphPlus creates for the purpose of ARexx interaction. The name of this port is:

```
ADPro
```

You can make the connection from ARexx to MorphPlus using the following ARexx command:

```
ADDRESS "ADPro"
```

ARexx will generate an error condition should MorphPlus not be found.

NOTE: The MorphPlus ARexx port name is case specific. It must be specified exactly as shown above, otherwise interaction will not be possible.

8.2 Getting Results of Commands

After every command directed at MorphPlus from ARexx is executed, MorphPlus will fill in the standard ARexx result variable, **RC**, with a return value. If this value is 0, then the previous command executed without error. If the value of **RC** is not zero, then it contains an error severity indicator.

In order to receive additional error indications or to receive non-error status information back from MorphPlus, you must request additional status information from ARexx using the following command:

```
OPTIONS RESULTS
```

This command must be one of the first executable instructions in your ARexx program.

If an **OPTIONS RESULTS** command has been issued, then MorphPlus will load additional information into the ARexx variable **ADPRO_RESULT**. **ADPRO_RESULT** can be consulted whenever **RC** is non-zero to gain additional information about the nature of an error.

When **RC** is returned with a 0 value (indicating no error occurred), **ADPRO_RESULT** will be set to command specific information. Note that this means you must check **RC** before depending on the data in **ADPRO_RESULT** since **ADPRO_RESULT** may contain either additional error information or the status information you requested.

The **ADPRO_RESULT** variable is available only to true ARexx programs. If you are sending ARexx style commands to MorphPlus from a non-ARexx program, then results which normally would be passed back in **ADPRO_RESULT** will instead be passed back in the **rm_Result2** field in the ARexx message structure.

The ARexx command specifications (in Chapter 12) usually list how each command affects the **RC** and **ADPRO_RESULT** variables. If either one or both is missing (or if the variable is labelled "unchanged"), then that particular command doesn't change the unlisted variable's value. If it is labelled "undefined", then

the value may or may not have changed. This label means you should not depend on the value of this variable.

8.3 Launching ARexx Programs

MorphPlus can launch up to 50 distinct but specially named ARexx programs. These programs must reside in your **REXX:** directory and must be specially named according to the following conventions:

- If any function key (**F1** through **F10**) is pressed, MorphPlus will attempt to launch an ARexx program named **F1.adpro** through **F0.adpro**. For example, if **F8** is pressed MorphPlus will attempt to launch **F8.adpro**.
- If any function key (**F1** through **F10**) is pressed along with either **Shift** key, MorphPlus will attempt to launch an ARexx program named **SF1.adpro** through **SF0.adpro**. For example, if **Shift-F10** is pressed, MorphPlus will attempt to launch **SF0.adpro**.
- If any function key (**F1** through **F10**) is pressed along with either **Alt** key, MorphPlus will attempt to launch an ARexx program named **LF1.adpro** through **LF0.adpro**. For example, if **Alt-F3** is pressed, MorphPlus will attempt to launch **LF3.adpro**.
- If any function key (**F1** through **F10**) is pressed along with either **Amiga** key, MorphPlus will attempt to launch an ARexx program named **AF1.adpro** through **AF0.adpro**. For example, if **Amiga-F6** is pressed, MorphPlus will attempt to launch **AF6.adpro**.
- If any function key (**F1** through **F10**) is pressed along with the **Ctrl** (Control) key, MorphPlus will attempt to launch an ARexx program named **CF1.adpro** through **CF0.adpro**. For example, if **Ctrl-F2** is pressed, MorphPlus will attempt to launch **CF2.adpro**.

If more than one of the **Ctrl**, **Shift**, **Alt**, or **Amiga** keys are depressed at the same time as a function key, MorphPlus will *not* attempt to launch an ARexx program.

If one of the key combinations (described above) is depressed for which there is no corresponding ARexx program, you will be told.

We provide an ARexx program attached to the **F10** function key which allows other ARexx programs to be launched by name. This program, called **F0.adpro**, will ask you to specify an arbitrary ARexx program to execute. This allows you to use ARexx programs which do not conform to the naming conventions listed above. The **F0.adpro** program comes on the MorphPlus distribution disk. You can install it into your **REXX:** directory by running the installation script and choosing the option that installs the sample ARexx programs.

8.4 Understanding the General Rules

Three general rules apply to all MorphPlus ARexx commands. These are:

1. All commands set `RC` and `ADPRO_RESULT` as indicated above.
2. Most commands which change some internal state in MorphPlus return the old value in `ADPRO_RESULT` if no error occurred. If a command which requires additional arguments to change an internal state is executed without the required arguments, the old value of the state is returned and is not changed. Therefore, a command which changes some state can also be used to interrogate the present value of the state without changing it.
3. All commands are case-insensitive. However, arguments may be case sensitive.

8.5 Using ARexx Commands

For a complete listing of all ARexx commands available in MorphPlus, please refer to the various sections in Chapter 12. These ARexx commands are grouped by subject matter—main screen functions, loaders, savers, and operators. Commands which do not belong to any of these groups are listed in Section 12.2.5 in Chapter 12.

Listed below is a complete cross reference of all ARexx commands available in MorphPlus, along with page references to their descriptions. The commands are listed in alphabetical order.

Command	Description	Page
ABS_SCALE	Scale the image to absolute dimensions.	253
ADPRO_DISPLAY	Display the rendered image.	199
ADPRO_EXIT	Exit the program.	201
ADPRO_TO_BACK	Move the screen behind all others.	202
ADPRO_TO_FRONT	Move the screen in front of all others.	201
ADPRO_UNDISPLAY	Remove the display of the rendered image.	200
BLUE	Set the blue content of the rendered image.	203
BRIGHTNESS	Set the brightness of the rendered image.	203
CONTRAST	Set the contrast of the rendered image.	204
CURRENT_MEM_SIZE	Get the current size of the image buffer.	201
DITHER	Set the dither method to use.	212
EXECUTE	Create rendered data from raw data.	199
GAMMA	Set the gamma correction for the image.	204
GETDIR	Ask the user to select a directory.	219
GETFILE	Ask the user to select a file.	217
GETFILES	Ask the user to select a list of files.	218
GETNUMBER	Ask the user to enter a number.	217
GETSTRING	Ask the user to enter a string.	216
GREEN	Set the green content of the image.	202
IMAGE_TYPE	Get the type of image data available.	215
LAST_LOADED_IMAGE	Get the filename of the last loaded image.	214
LAST_SAVED_IMAGE	Get the filename of the last saved image.	213
LFORMAT	Set the loader format.	224
LOAD	Load an image into MorphPlus.	224
OBTAIN_ADPRO	Obtain exclusive lock on MorphPlus.	221
OKAY1	Display a message with one button.	220
OKAY2	Display a message with two buttons.	220
OPERATOR	Execute an operator module.	237
ORIENTATION	Set the load orientation.	226
PAUSE	Cause a delay in the execution of the script.	216
PCONTRAST	Define whether colors 0 and 1 should contrast.	207
PCT_SCALE	Scale the image by percentages.	253
PGETWB	Load the first four Workbench colors.	209
PIXEL_ASPECT	Get the image's pixel aspect.	221
PIXEL_RESOLUTION	Get the image's pixel resolution.	221
PLOAD	Load a palette file.	209
POFFSET	Specify a register as offset color zero.	207
PPOKE	Set the value in a color register.	208
PSAVE	Save the contents of the palette to a file.	210
PSORT	Specify the sort ordering of the palette.	208

Command	Description	Page
PSTATUS	Specify whether to lock the palette or not.	205
PTOTAL	Set the total number of colors.	206
PUSED	Set the number of colors to actually use.	206
PWIDTH	Specify the accuracy of the palette.	205
RED	Set the red content of the image.	202
RELEASE_ADPRO	Release exclusive lock on MorphPlus.	222
RENDER_TYPE	Set the rendering mode.	199
SAVE	Save the image currently in MorphPlus.	234
SCREEN_TYPE	Set the type of screen to use.	211
SERIAL_NUMBER	Get the program's serial number.	200
SFORMAT	Set the saver format.	233
VERSION	Get the program's version number	200
XSIZE	Get the width of the image.	214
YSIZE	Get the height of the image.	214

8.6 Using the Supplied ARexx Programs

`f0.adpro`

Executing this program brings up the file requester in the `REXX:` directory allowing you to pick (by name) another ARexx program to execute. We find this program so useful, we bind it to your `F10` function key (by calling it `F0.adpro`).

Remember, you may have to modify this program so that it points correctly to your RX program. See the program for more information.

`getfiles_example.adpro`

This program is an example of using the `GETFILES` ARexx command. `GETFILES` is like `GETFILE` in that it brings up the file requester, but `GETFILES` allows the user to select multiple files at one time.

This ARexx program will ask you to select multiple files and then steps through your selections.

`last_loaded.adpro`

This program makes use of the `LAST_LOADED_IMAGE` command to tell you the name of the file you last loaded.

`locate-adpro.adpro`

This program is an example of how you might start MorphPlus from ARexx if you discovered it wasn't running at the time your own ARexx program began executing.

`scaletoaspect.adpro`

This program is an answer to those many people who wanted an automatic way to scale to a specific pixel aspect. As long as the pixel aspect of your image data is set correctly in the first place, this ARexx program figures out how to scale the image to the aspect you want.

For example, suppose you scanned an image (from a 1:1 pixel aspect scanner) and you want to scale the image for display in a hi-res interlaced screen. Simply tell this program you want a 22 to 26 image (when it asks you) and it handles the rest.

Chapter 9

Morph

Creating morphs with MorphPlus is accomplished with the stand-alone program called Morph. Morph allows you to transform one image into another across a user-specified number of frames. It communicates with the MorphPlus' Warp operator to create the individual frames of the morph.

NOTE: MorphPlus needs to be running at the same time for Morph to operate. If you do not have enough memory available, you will not be able to run Morph and MorphPlus simultaneously. We suggest you try running only these two programs in your system (and after "booting-up" the computer) if you find that Morph cannot open its control screen because of not enough available memory.

NOTE: Please read through the Warp operator description (Section 7.19) before continuing with this section. Only those areas which are different than those in the Warp operator, as well as those that are specific to Morph will be described below.

Start the Morph program by double-clicking on its icon. A separate grayscale screen will appear, as shown in Figure 9.1.

Morph utilizes a Visual User Interface, just like many of the operators. In fact, the interface has many things in common with the Warp operator.

To determine the version number of the Morph program, select the **About...** menu item (from the **Project** menu).

To exit Morph, select the **Quit** menu item. If you have any work that has not

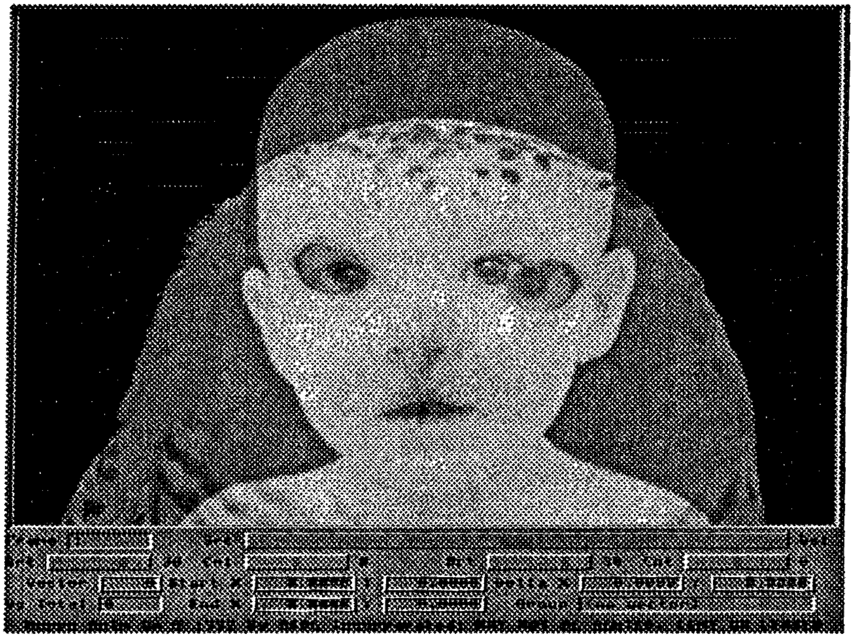


Figure 9.1: Components of the Morph AnimOp Screen.

been saved, you will have a chance to do so before the program terminates.

The first difference you will notice is that no image is automatically displayed on the screen. The images in Morph are handled a little bit differently. These differences are explained below.

9.1 Projects

All work done with the Morph program is through the concept of Projects. Think of your morphs, whether it be a morph between two static images or two sequences of moving images, as your "work" or "project". Just like any type of work you do, you can start a new project, work on old projects, preserve changes made to them, and discard them altogether. With Morph, you can do all of these things.

The first thing you need to do is create a Project for your morph, if one does not already exist. If you are morphing between two static (still) images, select the **Still Morph...** menu subitem under the **New** menu item (under the **Project** menu). The Still Morph control panel will appear (the second control panel in Figure 9.2), from which you can define the source and destination images and the name of the Project. This Project name will be used to differentiate one Project from the next.

If you are working on a moving morph (and have sequence files already generated for the source and destination sequences), select the **Moving Morph...** subitem. The Moving Morph control panel (the top control panel in Figure 9.2) will appear, but instead of asking for single images it will ask you to specify the source and destination sequences between which a morph will be created. If one sequence has fewer frames than the other, then you can pad the shorter sequence out to the length of the longer one (repeatedly using the last frame of the shorter one) by setting the **If Different Length** cycle gadget to **Repeat Last Frame Of Shorter Sequence**. Alternatively, you can cut both sequences to the shorter length by selecting **Clip To Length Of Shorter Sequence**.

The Morph program can also create warps of a single image (still warp) or sequence of images (moving warp). The difference between a warp and a morph is that warps work on one image at a time (be it a single image or a sequence of images), whereas morphs work on two images at a time. To create a still warp, select the **Still Warp...** menu item. Enter the name of the project by either clicking on the **Project Name...** button or typing it into the input field. Select the source image using the **Source Image...** button or input field. To create a moving warp, select the **Moving Warp...** menu item. Instead of entering a source image, you would enter a sequence (in the **Source Sequence...** input field).

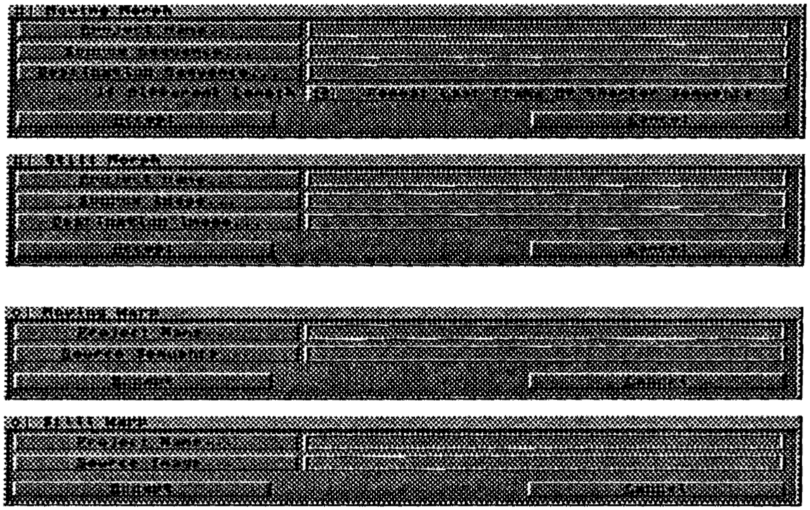


Figure 9.2: The New Project control panels. From top to bottom: Moving Morph, Still Morph, Moving Warp, Still Warp.

NOTE: Unless otherwise indicated, the description of a still or moving morph in this chapter will also apply to a still or moving warp.

To give you a graphical representation of these different types of projects, as well as how they relate to the Warp operator, see the diagrams in Figure 9.3.

When you press **Accept**, the Project Refresh control panel (as shown in Figure 9.4) will appear. The resolution in which the VUI images will be displayed is defined in this panel. The **GUI Image Resolution** can be toggled between **High Resolution** and **Low Resolution**. Using **High Resolution** allows you to see more detail in the image at the expense of disk space, whereas **Low Resolution** gives you a rougher representation of the image but doesn't require as much disk space. If you are doing a quick morph that doesn't require too much detailed placement of vectors or if you cannot afford the space on disk, use **Low Resolution**, otherwise **High Resolution** should be your choice. If you switch from one resolution to the other, Morph will have to update the files stored in this project's directory, which may take some time to complete.

The three radio buttons labelled **Generate All**, **Generate Source Only**, and **Generate Destination Only** control which VUI images (see Section 9.2) of a morph (still or moving) will be generated. For most cases, you can just leave the **Generate All** button selected. If you are low on hard disk space, then you can switch to one of the other two choices, although moving from frame

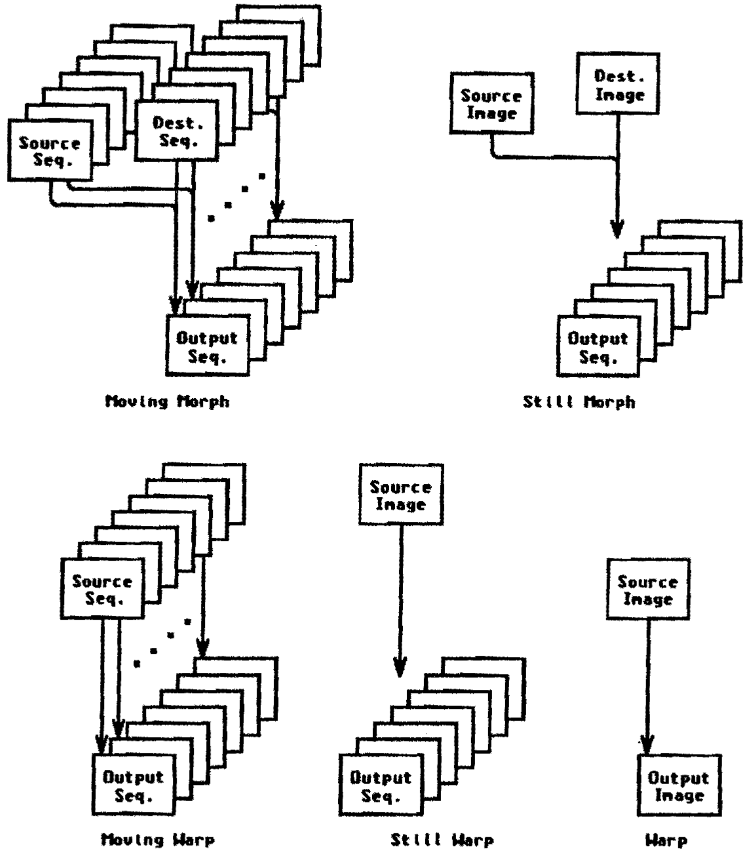


Figure 9.3: The Morph vs. Warp Operations.

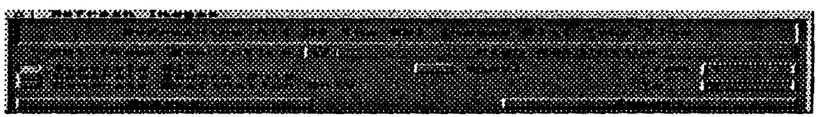


Figure 9.4: The Refresh Images control panel.

to frame might take longer.

If many people are working on a project but you want all of your VUI images to always be up-to-date, make sure the **Smart** checkbox is selected (checked). When checked, if any image file in the current project's directory has been updated, that particular image will be reprocessed so that the image you see on screen is the most recent. This feature compares the date stamp (the day and time of last modification) of the files, so be sure that your system clock is properly set. You can use this feature as many times as you want, so that you can be assured of working on the newest versions.

The **From** and **To** input fields allow specify—for a moving warp or morph—the first and last frames to refresh.

If you select **Accept**, a directory will be created using the name you specified for the project. This directory contains the following items:

- 1 (still) or 2 (moving) sequence file(s)
- 1 (still) or 2 (moving) image file(s) per frame; these are used for the onion skin VUI, and are not an unmodified copy of the original images.
- 1 vector file per frame
- 1 group file
- 1 preferences file

The process of creating new Projects may take some time to complete and will use as much disk space required to store these intermediate files. Once a Project has been defined, you will be able to begin creating the morph or warp.

If you had previously created a Project, you can resume working with it by selecting the **Open...** menu item (under the **Project** menu) to open a previously saved Project. A control panel will appear, from which you can manually enter the name of the project in the input field. Alternatively, you can press the **Select Project...** button and a file requester will appear, showing a list of directory names. Recall that when you create a new Project, Morph creates a directory with that name.

Any work that you have done on your Project can be saved at any time with the **Save** menu item. **Save** will record any changes you have made to the current Project under its current Project name.

You can also close the current Project by selecting the **Close** menu item. If you made any changes to the Project, you will have a chance to save them before closing the Project. If you do not save them, any changes you made since opening the Project will be lost. Closing a Project effectively disables much of the functionality in Morph.

If you do not want a Project anymore, you **must** use the **Delete...** menu item to remove it, and **not** use a directory utility to delete the files. You will be asked to confirm this operation—once a Project is deleted, it cannot be undeleted.

NOTE: Morph generates a few temporary files and places them in the directory of your choice (see Section 9.7). The initial directory that will be used is the `T:` *assigned directory (which is, by default, assigned to the* `RAM:T` *directory). If you do not want to place these temporary files in* `RAM:`*, change the directory specified in the* **Temporary Directory...** *input field to a different location—preferably somewhere on your hard disk (if you have one).*

9.2 Onion Skin VUI

The Morph user interface lets you work on both images at the same time, and not just one image at a time. Morph uses an "onion skin" visual user interface (VUI) where the source image can be mixed with the destination image, making the images seem transparent. This allows you to easily and accurately move start and end points between source and destination images, reducing the set-up time required.

The VUI can display a fairly large number of "skins", where each "skin" represent a layer of the source image mixed with the destination image for a given frame. The topmost skin layer represents 100% of the source image and 0% of the destination image, meaning only the source image is visible. The bottommost layer is 0% source and 100% destination. All the layers inbetween (from the top layer to the bottom) represent decreasing percentages of source image being mixed with increasing percentages of destination image. The number of skins is represented by a unique knob position on the slider gadget. This number is large enough so that you will get subtle changes in source/destination mixing between any two neighboring knob positions.

To display a particular skin, use the slider gadget at the bottom of the screen. The knob's position within the gadget corresponds to the "depth". To the left of the slider is the word **Source**. If the knob is at this location, only the source (Start) image is displayed. The right side of the slider corresponds to the **Destination** (End) image. If the knob is in the middle (Middle) position, 50% of the source image is displayed with 50% of the destination image. Each knob position corresponds to a specific percent decrease of source image with an equal percent increase in destination image. Try sliding the knob slowly from side to side and you will see this mixing happen in real-time. Please note that these layers are generated "on the fly", but the refresh time should be reasonable.

Two keyboard shortcuts are available for moving this slider knob. The `c` and `v` keys will move the knob one position to the left or right, respectively. To move the know in bigger steps, hold down the `Shift` key while pressing either key. To move in even bigger steps, use the `Alt` key. To move to the leftmost or rightmost position, which will display all source or all destination, use the `Ctrl` key. These shortcuts allow you to keep your hands on the keyboard while you're creating and positioning vectors.

In the Warp operator's control screen, there is one pair of slider gadgets for brightness and contrast. In Morph, two are given. The left pair of **Brt** and **Cnt** sliders control the source image, while the right pair controls the destination image. These controls work the same way as in Warp, but they will be more useful here because you can control the brightness and contrast of a particular image without having to move the onion skin VUI slider to get a similar effect.

NOTE: For some images, changing the brightness or contrast may slightly modify the location of certain features in an image (such as the border between the lower lip and chin of a face). If you want your vectors to be properly placed, set the brightness and contrast controls, for both source and destination images, to the desired levels before *you begin defining vectors and points.*

9.3 Frames

The term "frame" describes the current pair of images, one image taken from the source sequence and the other taken from the destination sequence. For a morph between two single images, there is only one frame available. Remember that for single image morphs, Morph creates a source and destination sequence where each sequence contains just one image. For a moving morph (a morph between two sequences of images), the number of frames is equal to the number of frames in either sequence.

The current frame you are modifying is noted at the bottom of the screen. The vectors and edges that morph this frame's source image into the destination image are displayed on the grayscale images themselves.

If you want to work on another frame of the morph, you can use the **Previous**, **Next**, or **Specify...** menu items. **Previous** and **Next**, like their names suggest, allow you to move to the previous or next frames, respectively. The **Specify...** menu item will display the Specify Frame control panel (shown in Figure 9.5, from which you can enter the number of the frame to "jump to".

Try not to get the terms "onion skin" and "frame" mixed up. A frame describes a morph between one image in the source sequence to the corresponding image in the destination sequence. Onion skin refers to a specific frame's visual

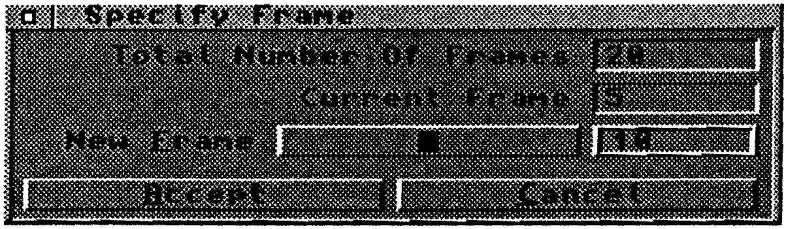


Figure 9.5: The Specify Frame control panel.

representation (percentage of source image that is mixed with the destination image).

9.4 Vectors

Vectors behave similarly as in the Warp operator, although you have more control over the affect they will have on the images in Morph. With the Warp vectors you are only modifying a single image, whereas in Morph you are modifying two of them. Before we discuss how vectors are described differently in Morph, you should know how the morphing will affect the two images.

When you are morphing an object in one image into an object in a second image, you are describing the relationship or correspondence between points on the first and second images. For example, if you were to morph a circle into a square, you have to specify where certain key points on the circle will move to so that they, after a specified number of frames, transform into a square. As each point is tracked across generated frames in the morph, they contribute less to depicting a circle and more to depicting a square.

Another way of stating this is that near the beginning of the morph the start points move a small distance (relative to the total distance they have to "travel") while end points move a large distance. If this doesn't sounds right to you, you are not alone. This can be a very difficult concept to comprehend. Each point in the source image has to end up at a possibly different point on the destination image. That is, each point, at any instance during the morph, moves a fraction of the way to its final location. For example, if the start point moves 25% of the distance, the corresponding end point moves 75% (for a total of 100%) to "meet up with it".

A very useful feature is the ability to specify which image a particular vector will affect. A vector can affect both the source and destination images, only the source, or only the destination.

A source and destination vector correlates the selected areas between source and destination images so that they match up correctly. This is the type of vectors which you will probably use most often. These vectors are visually defined by the use of square (start and end) points. To make a vector a source and destination vector, click on the vectors you want to modify and select the **Common To Both** menu item (in the **Vectors** menu). If the vector's points were not previously square, they will be.

A source only vector affects the source image independent of the destination image. This allows you to warp the source image without warping the corresponding areas in the destination image. To make vectors become source only, click on the vectors you want to change and select the **Source Only** menu subitem (under the **Vector Type** menu item). Each of the selected vectors will have its points change to triangles that point to the right side of the screen. This should give you a visual cue that these vectors only affect the source image because to move from source to destination image, the slider knob has to move to the right.

A destination only vector affects the destination image independent of the source image. To make vectors become destination only, click on the vectors you want to change and select the **Destination Only** menu item. The selected vectors' points turn into triangles pointing to the left side of the screen, which cues you to the direction from destination to source.

For example, if your destination image is a picture of a face (with an open mouth) and you want its mouth (starting with a closed mouth) to open over time, you would place your destination only vectors' start points (filled squares) in the "closed mouth" position and the end pints (hollow squares) outlining the "opened mouth". At frame 1 (start of the morph), the mouth will be closed since the start points define the shape of a closed mouth. As the morph progresses, the mouth will start to open. At the last frame (end of the morph), the mouth will be fully open.

If this example of destination only vectors seemed hard to understand, don't be discouraged. Another way of using destination only vectors is to think of the morph in reverse—the mouth will close. If you think of it this way and place destination only vectors with the start points outlining the opened mouth and end points at the closed mouth positions, then as a final step use the **Flip Selected** menu item, you will end up with the proper orientation of start and end points.

NOTE: We suggest you use source only and destination only vectors sparingly. If you do decide to use them, you should surround the particular vectors with points (including edges between these points) to minimize the affect that single-image vectors will have on areas of the image that also have common (source and destination) vectors. If you don't do this step, you might create an undesirable effect.

In addition to what was discussed in the Warp section regarding vector visibility, the Morph program includes three additional menu subitems in the **Visibility** menu item. **Hide Source Only** will hide all source only vectors, **Hide Destination Only** will hide destination only vectors, and **Hide Common** will hide common vectors (those that work on both source and destination images).

Three additional keyboard shortcuts are available for moving the slider knob. The `d` key will move the slider to the leftmost position (showing all of the source image) and display only the vectors' start points. The `g` key will move the slider to the rightmost position (showing all of the destination image) and display the end points. These two keys are useful for seeing the positions of your vectors in both beginning and ending images, without you having to reach for the mouse and manually slide the knob and choose a menu item. The third key, `f`, will display a 50/50 mix of source and destination images, displaying both start and end points of the vectors, and placing the knob at the center of the slider.

Consult the Morph program's reference section (Section 12.6) for not only a summary of all controls, but also a listing of all keyboard shortcuts.

9.5 Tweening

For moving morphs or warps, the process of creating vectors for each frame is very time-consuming...especially when the location and size of these vectors may not change much from frame to frame. Morph's group tweening feature significantly reduces the set-up time required by automatically generating the vectors (on a group-by-group basis) for a range of frames.

Select the **Tween...** menu item (in the **Groups** menu) to bring up the Group Vector Tweening control panel (as shown in Figure 9.6). As you can see, this panel looks similar to the Delete Groups and Group Visibility control panels. In this particular panel, the groups shown in the left hand list (**Don't Tween**) will not be processed (tweened), while those that are moved to the right hand list (**Tweened Groups**) will be.

To move a group from one list to the other, select the group name and press either the left arrow or right arrow "move" button, depending upon the di-

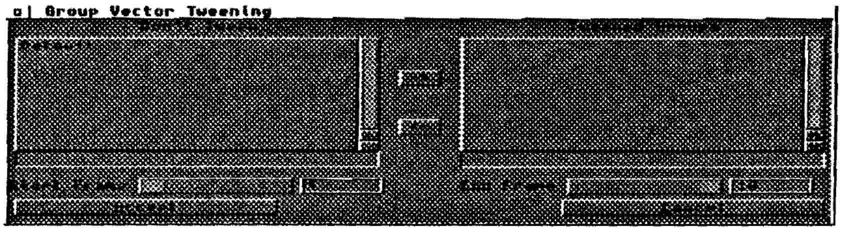


Figure 9.6: The Group Vector Tweening control panel.

rection it needs to "move." A simpler way is to use the double-click method described in the Warp operator (Section 7.19.4 to be more specific).

To specify the range of frames over which all selected (right hand list) groups will be tweened, change the **Start Frame** slider value to a number between 1 and one less than the number of frames currently defined in the sequence. Similarly, the ending frame is defined by the **End Frame** value (between 2 and the number of frames).

Once you press the **Accept** button, all frames between the two selected will have a new set of vectors generated. These vectors will correspond to each group's vectors, as they gradually transform from the start frame's definition to the end frame's. Pressing the **Cancel** button exits from the control panel without any tweening being done.

NOTE: Please take caution when using this feature, as it can be easy to generate a set of duplicate vectors for a frame. Morph has no way of knowing, for a given frame, whether the vectors for a given area of the moving image(s) have already been defined. For example, if you perform the tween operation twice on the same set of groups over the same range of frames, the generated vectors will effectively cancel themselves out; Morph as described in Section 7.19.3, will filter out vectors whose start and/or end points are the same.

9.6 Preview Animation

Another useful feature (for moving morphs or warps only) is the ability to preview the location and movement of the defined vectors.

Select the **Animate...** menu item (from the **Groups** menu) to display the Group Vector Animation control panel (shown in Figure 9.7). As with most other group manipulation control panels, this one uses the "two list" method.

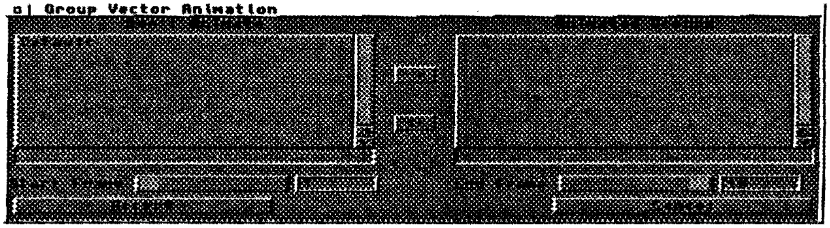


Figure 9.7: The Group Vector Animation control panel.

The entries in the left list (**Don't Animate**) will not be animated, whereas the entries in the right list (**Animated Groups**) will.

Once you press **Accept**, a separate screen will appear, showing the movement of the selected groups' vectors over the selected range of frames. Only the vectors themselves, and not the actual images, will be displayed in this animated preview. This allows you to get a real-time preview of the motion of your vectors over "time" (the specified range of frames). Making small changes to your vectors or to the motion curves may introduce inconsistencies that you are trying to avoid. Using the group animation feature allows you to see what a particular change will do to the motion.

Pressing **Cancel** aborts the operation, returning control to the main Morph screen.

9.7 Preferences

The ability to customize your working environment can make your current job at hand go faster and smoother. Morph allows you to set various parts of its environment with the Preferences control panel. Select the **Preferences...** menu item to display this panel (shown in Figure 9.8). As you can see in this picture, the **Colors** controls are the same as in the Warp operator's Preferences panel. This section will only describe those controls that are specific to the Morph program.

When a file requester is displayed, it displays the contents of the current directory. If you want to customize the directories that are displayed, you may do so with the four provided buttons and input fields. **Image Directory...** lets you specify the default directory that Morph will display when it asks for you to specify an image (for still morphs or warps). The **Project Directory...** defines the default directory for projects. **Sequence Directory...** is for sequence files (moving morphs and warps). The **Temporary Directory...**

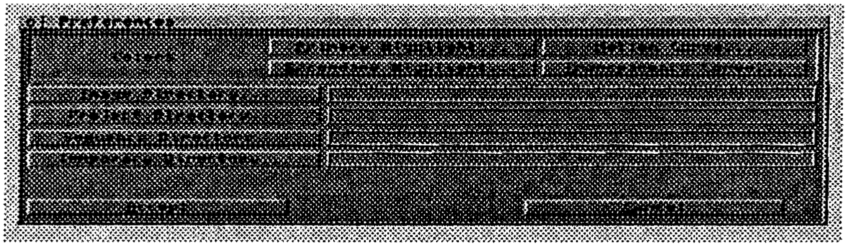


Figure 9.8: The Morph Preferences control panel.

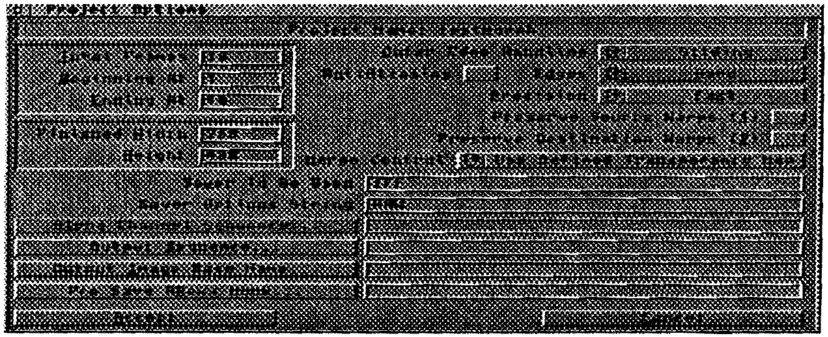


Figure 9.9: The Project Options control panel.

buttons specifies the directory that will contains some temporary files that Morph creates and uses during the execution of the program and processing of a morph or warp.

Press the **Accept** button to retain and use the current Preferences settings, or press **Cancel** to preserve the previous settings.

9.8 Project Options

Each project you create has several options that can be defined, including which images should be generated, the location of these images, the type of morphing or warping technique to use, how edges should be handled, and the precision of the operation. Select the **Options...** menu item from the **Project** menu to display the Project Options control panel (as shown in Figure 9.9). It is from this panel that you set the global controls for the current project.

9.8.1 Merge Control

For morphs, each pre-morph frame (two of them for a moving morph, one for a still morph) can be independently preserved with the use of the **Preserve Source Warps (1)** and **Preserve Destination Warps (2)** check boxes. If either option is chosen, then that corresponding image, after being used to create the resulting frame, will be saved and not discarded.

The process of merging these two warped images is controlled with the **Merge Control** cycle gadget. If set to **Don't Perform Merge**, then the actual compositing of the source and destination images will not be performed. Probably the only reason you would want to do this is if you were interested only in the individual warps (and not the resulting image).

To use the group-specific transparency settings (see Section 7.19.4 for details), make sure this gadget reads **Use Defined Transparency Map**.

You can override these settings by setting **Merge Control** to another value. Your group information is not lost—just not used.

The **Use Linear Transparency Map** option makes all group transparency curves linear (a straight line going from **start** to **end**). This basically creates a smooth and consistent transition between source and destination images.

NOTE: If you have not modified any of your groups' motion and transparency curves from their default (linear) shape, you should use the **Use Linear Transparency Map** *setting rather than* **Use Defined Transparency Map**. *Even though the curves are shaped the same, the* **Use Linear Transparency Map** *choice will make the morph operation finish faster than* **Use Defined Transparency Map**.

If you want all of the non-Default groups to use the Default group's transparency curves, then set the cycle gadget to **Use Default Transparency Map**. All non-Defaults groups will be able to use the Default's transparency curve, without you having to manually change each of them. Actually, the current curve definitions are not modified at all, so you'll be able to try out different merge settings while still preserving any custom curve definitions you may have set.

The **Use 50% Transparency Map** option allows you to set all transparency curves to a 50% level (50% of source image mixed with 50% of destination, across all frames). This is very handy for when you want to create a preview of the images.

The **Use External Alpha Channel** option lets you use a sequence of grayscale IFF images (referenced with a sequence file) as the transparency control for each frame. If this option is selected, the **Alpha Channel Se-**

quence... button and input field will be enabled so that a valid sequence file can be entered. A few restrictions are made on this sequence of images:

- all images must be in grayscale IFF format
- all images must be of the same dimensions (width and height)
- the number of images in this sequence must be equal to the number of images in both the source and destination sequences.

NOTE: Even if the selected alpha channel sequence doesn't follow these rules, Morph will still try to use it. If if comes across a frame that breaks one of these rules, the operation will fail at that point. The alpha channel sequence images are **not** *checked for correctness before the morphing operation begins, so be sure that you are specifying a proper sequence.*

9.8.2 Output Control

The quality or precision of the warping can be controlled with the **Precision** cycle gadget setting. **Fast** will perform the warp as quick as possible, use less memory, but sacrifice some quality. Conversely, **High Quality** produces a higher quality, but will use more memory and may take longer to complete.

The generated frames of the morph or warp are saved with the file base name specified in the **Output Image Base Name...** input field. A sequence file (usable by Morph or FRED) can also be generated by specifying its name in the **Output Sequence...** input field. A handy feature is the ability to process each generated frame before having Morph save it out to disk. You can write an ARexx script that uses other image processing functions of MorphPlus to "touch-up" the image before being saved. Enter the name of this (optional) ARexx script in the **Pre Save ARexx Hook...** input field.

NOTE: All sequence files used by Morph and FRED require the "`.seq`*" filename extension. Without it, the files will not be recognized by either program as a valid sequence file.*

The format and image data type of these frames can be specified in the **Saver To Be Used** and **Saver Options String** input fields. The **Saver To Be Used** is the name of the saver, the same spelling as shown on the **Save Format** button and used with the `SFORMAT` command through ARexx. The **Saver Options String** are the arguments that follow the *filename* argument on the `SAVE` command line (through ARexx).

For moving morphs and warps, the total number of frames that will be generated is displayed in the **Total Frames** text field. For still morphs and warps,

this number can be changed (shown in an input field) to specify how many frames will be calculated. For all cases, the **Beginning At** and **Ending At** input fields let you specify the first and last frames to actually generate. This allows you to produce selective frames, without having to create the entire sequence of frames.

The dimensions of the resulting images can be defined with the **Finished Width** and **Height** input fields. Note that each pre-morph or pre-warp image will be scaled to this dimension *before* being processed.

Once you are satisfied with the options, press the **Accept** button. **Cancel** will exit the control panel and restore the previous values.

9.9 Other Options

All the files that can be saved from Morph can have icons attached to them. These include projects, vector files, group files, preferences files, and sequence and single image files. If the **Create Icons** menu item is checked, then icons will be attached to these files (only those created after the menu item was checked).

All of the settings in the Morph program can also be stored and retrieved, allowing for custom work environments. This can be handy when you are working on many projects at the same time, with each requiring different options. The **Load Settings...** menu item will use the settings defined in a previously-saved Morph settings file; **Save Settings** will store the current settings under the default settings filename; **Save Settings As...** will store the current settings with a user-specified settings filename.

NOTE: Although most of MorphPlus' settings are saved in the `ADProDefaults` *file when you exit that program, Morph's settings will only be saved if you tell it (using either the* **Save Settings** *or* **Save Settings As...** *menu items). The default program settings are saved in a file called* `Morph.pref`*, located in the same directory as the Morph program.*

To make it easier for you to jump from one program to the next, Morph has a few menu items (located in the **Applications** menu) that let you move to the MorphPlus/ADPro, FRED, or Workbench screen from the Morph screen. Select the item—**MorphPlus/ADPro**, **FRED**, or **Workbench**—corresponding to the screen you want to display in front of other screens. If you select either **MorphPlus/ADPro** or **FRED** but it is not currently running in the background, Morph will make every effort to start the program. It will use the `ADPRO:` and `ADP_FRED:` assignments to locate the programs.

Chapter 10

FRED

FRED is a visually oriented list manager. The lists it manages, called sequences, are comprised of image names and other information. Individual images (or frames) in a sequence are represented on-screen with accurately rendered icons.

FRED is an independently executable program which can be run in conjunction with ADPro.

FRED can instruct ADPro to process each frame in a sequence, providing an easy method of batch processing. FRED also provides an extensible ability to call special purpose drivers which can cause ADPro to generate animation effects automatically. You can even preview animations within FRED, before (for example) committing them to a single frame recorder.

FRED is a powerful tool for the 24 bit single animator. It also benefits those who would like to batch process unrelated pictures but feel they are not up to the task of writing the appropriate ARexx program.

Note that FRED, just as the rest of the MorphPlus package, requires Kickstart/Workbench 2.0 or higher of Commodore's Amiga operating system. FRED also requires that ADPro be running at the same time for it to run. To start FRED, double-click on its icon. A separate control screen will appear.

Before we describe the features of FRED, you should know how to exit the program. The **Quit** menu item will do this for you.

To find out which version of FRED you are running, select the **About...** menu item. A window will appear with this information, as well as a legal notice.

Two sets of menus are accessible from within FRED. When the backdrop screen

Depth	Colors
4	16
5	32
6	64
7	128
8	256

Table 10.1: Screen depths supported by FRED.

is active, a menu set including **Project**, **Options**, and **AnimOps** items can be accessed. When a sequence window is active, a menu set including **Project**, **Edit**, and **Miscellaneous** items can be accessed.

In both sets, the **Project** menu is the same.

The **Options** menu is available when the backdrop window is selected. It contains commands which allow you to set various options governing FRED's operation.

For instance, the **Screen Depth** menu item contains subitems that allow you to set the depth of the screen which FRED will use. FRED supports the screen depths shown in Table 10.1

The only depth which is supported by the hardware in most Amigas is **4 Bit-Planes**. New Amiga models installed with the Advanced Graphics Architecture (AGA) chip set, such as the Amiga 4000, support the other depths. If you are running FRED on one of these machines, then these menu items will be selectable. Note, the depth of the screen selected affects the amount of memory needed to display stamps in a sequence.

10.1 Sequences, Cells, and Frames

Sequences, as mentioned earlier, are lists of images that will be processed by FRED. These sequences can be assembled, loaded, saved, processed, and animated.

Creating a new sequence is as easy as selecting the **New...** menu item (from the **Project** menu). An empty window, called the sequence window, will appear. Its title bar displays the name of the sequence (initially defined as `UNNAMED`*xxx*`.seq`, where *xxx* is a number from 000 to 999), as well as if the sequence was modified at all (`[*]` for a modified sequence, `[ ]` if not modified).

These sequences are made up of cells (single images). These images can be

similar to one another (such as cells of an animation) or a collection of varied images. What they share in common is the need to be processed.

Images can be inserted into a sequence as follows. Click on the sequence window to add to; the window should now be active. Select either the **Insert Image(s)...** or **Insert Range...** menu item. **Insert Image(s)...** lets you choose one or more images (from the same directory) to insert into the currently active sequence. **Insert Range...** allows you to insert a group of images that have numeric file extensions. For example, you can select to load `mypic.0004` (the first image in the range) through `mypic.0020` (the last image) and FRED will automatically search for files with a base name of `mypic` and extensions of `.0004`, `.0005`, all the way up to `.0020`. Note that the selected images will be in inserted *after* the currently selected cell, which has its cell information box depressed and its text displayed in white.

If you want to save the defined cells of the sequence, select either the **Save** or **Save As...** menu item. **Save** will store the cells under its current sequence name (as shown in the window's title bar), if it has one. **Save As...** will let you save the sequence under a new name.

If no window is currently active (i.e., the backdrop screen is active), a list of currently open sequences will appear. If you select a sequence name and then press the **Accept** button, that sequence will be saved.

To open a previously saved sequence, select the **Open...** menu item and select the sequence to load from the file requester that will appear.

You can have many sequence windows open on the FRED screen as graphics memory will allow. These windows can be hidden from view (also termed "iconified") by clicking on a window's zoom gadget.

To close a sequence window, select the window and choose the **closE...** menu item.

To make a window reappear or to activate another open window, select the **Switch To...** menu item. A list of currently open sequences will be displayed. If an 'I' appears before a sequence's name, the window corresponding to that sequence is iconified. Select the sequence to switch to and press the **Accept** button. You can also double-click on the sequence name to bypass the **Accept** button altogether.

10.1.1 Cells vs. Frames

An important distinction in terminology must be made at this point before you continue reading the rest of the FRED documentation.

So far, we have termed the individual images of a sequence as "cells". What

hasn't been discussed (but what will be later) is that each cell does not have to be a single instance of an image—it can represent the same image as if it had been entered as successive cells, *without having to insert them.* This is accomplished through the use of "frames".

For example, say that you want a particular cell to represent five instances of an image. Instead of adding five cells to the sequence, you can add just one and specify a length (in the **Length** input field of the Image Info control panel) of 5. This ability comes in handy when animating the selected cells and batch processing the images.

10.2 Selection and Arrangement of Cells

Any cell in a sequence can either be selected or unselected. Any number of cells can be in the selected range. The currently selected (active) cell has a depressed information box with its text shown in white, whereas the cells in the current range have the background of their text (in their information boxes) shown in the highlight color.

To select a cell as the current one, simple click on it. Its information box will be depressed and its text shown in white. If you want a particular cell to be placed in the selected range, hold down the `Ctrl` key when clicking on the cell. The background of the text will become green, but the text color will stay the same (black). To remove a cell from the range, do the same operation.

You can also select a specific range of cells by clicking on the first cell in the range, selecting the **Select Range** menu item, and clicking on the last cell in the range. All of the cells between, and including, these two will be highlighted.

NOTE: When FRED works on the selected range, it will use the highlighted cells as well as the current cell, even if the current cell's text is not highlighted. *This is an important point to remember.*

You can also select all of the cells in a sequence by issuing the **Select All** menu item. Deselecting all of the cells is just as easy with the **Deselect All** menu item.

Cells do not have to stay in the same arrangement that they were added or inserted into the sequence—you can use the Clipboard to cut and copy cells from one part of the sequence and paste them in other parts. Note that only the information describing the cell, and **not** the image itself, will be posted to or retrieved from the Clipboard.

To remove a set of cells, select the cells and choose the **Cut** menu item. If you just want to copy the cells, then select **Copy** instead.

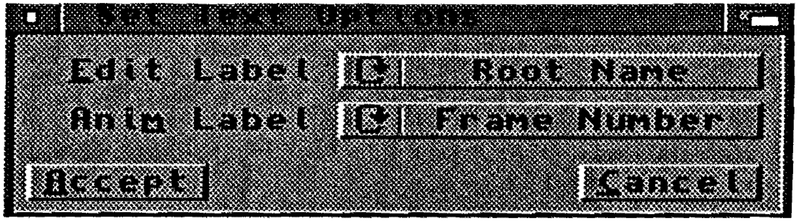


Figure 10.1: The Set Text Options control panel.

Pasting cells is done by selecting the cell that will preceed the pasted information and selecting the **Paste** menu item. If no cell is currently selected, the pasted cells will be appended to the end of the sequence.

You can also paste the cells in reverse order that they were placed into the Clipboard by selecting the **Reverse Paste** menu item.

NOTE: All cells that you cut or copy then later paste (using either paste command) keep their images' filenames. This is very important if you are going to process these images and overwrite the original version with the processed ones.

10.3 Cell Information

The **Set Text Options...** menu item will bring up the Set Text Options control panel (shown in Figure 10.1), from which you can change the text which appears below each stamp in a sequence as well as what text will be displayed during animation previews.

Both cycle gadgets (**Edit Label** and **Anim Label**) will let you select one of the following:

Root Name — This is the root filename of the image, as it can be found on disk. This filename does not include any path information.

Frame Number — A cell's frame number is a count of the number of "recordable units" which come before this cell. The first frame (first cell) in a sequence is frame zero. Each cell's length determines how many recordable units each frame occupies.

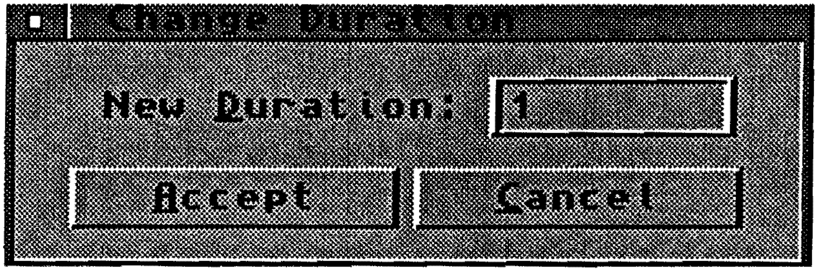


Figure 10.2: The Change Duration control panel.

For example, if a sequence contained just two cells each having a length of 1, then the frame number fields will show "00000000" and "00000001". If the first cell had a length of 60, then the cell's first frame number would still show "00000000" but the second frame number would show "00000060".

Time This shows the exact time at which the image will be displayed in the animation. The computation of time is based upon the current animation frame rate.

Time is expressed as minutes, seconds, and frames (or mm:ss:ff). A frame is a fraction of a second, whose duration is determined by the currently selected frame rate.

Length This shows the number of recordable units in the animation that the current stamp will be displayed.

Size This is the width and height (in pixels) of the image. This information is available only if a stamp has been created for this image. "Stamp" is another word for the icon which FRED uses to represent each picture.

The **Change Duration...** menu item displays the Change Duration control panel (shown in Figure 10.2) from which you can change the number of recordable units the currently selected cells will occupy in the animation. All cells have a default time of 1.

The amount of time a given cell's frames will actually be displayed depends upon the currently specified frame rate. For example, if the current playback speed was 30 frames per second, an image having a time of 15 will be displayed for half a second when the animation is played.

This Change Duration value can also be entered in the Image Info control panel (shown in Figure 10.3). To bring up this panel, either double-click on

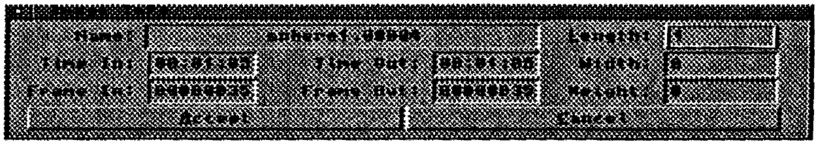


Figure 10.3: The Image Info control panel.

the cell itself, or single-click on the cell and either select the **Image Info...** menu item or press the RETURN key. In the Image Info control panel, you will see the filename of the cell, its currently defined length, the **Time In:** and **Time Out:** values, the **Frame In:** and **Frame Out:** values, and its **Width:** and **Height:**.

The **Time In:** and **Time Out:** values specify the display time (in seconds) for this cell. It takes into account the current **Animate Speed** (see Section 10.4) you have specified. For example, say that you have specified a speed of **15 Frames Per Second** and that the first three cells are of length 10, 5, and 20, respectively. The time in/out (frame in/out) values for these frames would be:

Time		Frame	
In	Out	In	Out
00000000	00000009	00:00:00	00:00:09
00000010	00000014	00:00:10	00:00:14
00000015	00000034	00:01:00	00:02:04

The **Frame In:** and **Frame Out:** values specify the first and last frame number described by this cell.

10.4 Animated Stamps

FRED allows you to preview your sequence of images in small, stamp-size images. You can specify the type of animated preview to produce, as well as the FPS (frames per second).

To create stamps for the currently selected cells, select the **Make Stamp** menu item. ADPro must be currently running for this operation to work. Each stamp will be placed in the same directory as the image it is making a stamp for, using a file extension of "**.stp**" to denote that it is a stamp.

Once all of the stamps have been created, you can animate them by selecting various settings and operations in the **Miscellaneous** menu.

The **Miscellaneous** Menu contains commands which allow you to set the frame rate (which affects how time is computed elsewhere), preview animations, create the stamps viewed in FRED, and cause generalized batch processing to take place. This menu is accessible only when a sequence window is selected.

You can choose to animate the entire sequence or the currently selected range by choosing either **Entire Sequence** or **Selected Range**.

The speed of the animation playback can be defined by choosing the desired frame rate from the **Animate Speed** menu item's subitems. Thirty FPS (frames per second) is the standard for NTSC video playback. Twelve FPS is useful for previewing animations which will be displayed on Commodore's CDTV. If FRED is running on a PAL machine, the options will be 25, 15, 12.5, and 10 FPS.

The setting you choose here affects how time is calculated elsewhere in FRED. If you set 30 FPS, for example, FRED will increment the seconds counter (in the time code display) once each 30 frames. Setting the frame rate to 10 causes the seconds counter to increment every 10 frames.

To actually animate the frames using the current animation settings, select one of the three types of motion: **Play Once**, **Continuous**, and **Ping Pong**. **Play Once** will play the frames only once. **Continuous** will play them over and over again, continuously. The last frame is followed by the first frame. **Ping Pong** is similar to **Continuous**, except that the frame order reverses upon displaying the last frame.

A small window will appear, showing the animated frames. If you find that the frames are flickering, move the window to the bottom of the screen. This should reduce this problem. To stop the animation, click on its close gadget.

10.5 The ADPro-FRED Link

NOTE: To process images with FRED and ADPro, you must be knowledgeable in writing simple ARexx scripts. Consult the ADPro manual (and even Chapter 8 in this manual) for more information on what needs to be included in a FRED ARexx script.

Your ARexx programs needs not concern themselves with the details of loading the image to be worked upon, nor do they need to know how to pick the next image to be processed. These tasks are performed by FRED before calling the first ARexx program in the chain you specify.

Arguments are passed to each ARexx program which indicate a frame's attributes. Use the ARexx `ARG` command as follows to fetch these values:

`ARG FrameNum FrameFName Length LoadFlag`

`FrameNum` will contain the current frame number (starting from 0). `FrameFName` will contain the current frame's filename. `Length` will contain the length (number of frames) of the current cell. `LoadFlag` will contain either 1 if the current image was loaded or 0 if the wasn't. This lets you determine the setting of the **Load** cycle gadget in the Invoke ADPro control panel. You can use these variables in your ARexx script to display meaningful error messages and customize filenames used with the `SAVE` command.

FRED allows you to process either all of the images in a sequence or a select few. Note that as with the other commands that reference the currently selected cells, the currently selected cell (shown with its text in white) is part of the selected group, even if the background of its text is not highlighted (green).

Choose the images you want to process and select the **Invoke ADPro...** menu item. The FRED Script List control panel (shown in Figure 10.4) will appear with the currently defined list of ARexx scripts. Notice that you can create a chain (list) of scripts (with the **Add...**, **Insert...**, and **Delete** buttons) that will be executed for each selected image in the current sequence. This allows you to assemble a complicated sequence of operations with smaller, more simpler, and more generalized scripts which you can reuse.

The buttons and cycle gadgets in this control panel are described below:

Add... A file requester will appear. Use it to enter the name of the ARexx script you would like to add. The selected ARexx program is added to the end of the chain.

Insert... This command allows you to add an ARexx script into the middle of the chain. First select the place in the chain where you would like to insert the script, then click on the **Insert...** button. A file requester will then appear in which you can enter the script name you would like to insert.

Delete This allows you to delete a script from the chain. First select the name of the script to be deleted, then click on the **Delete** button to remove the script from the chain.

Invoke This cycles between **Once Per Frame** and **Once Per Cell**, indicating when to invoke the list of scripts.

Load This cycles between **Every Frame** and **Once Per Cell**, indicating when to load each image.

Once you are satisfied with the list of scripts and are ready to process the selected images, press the **Accept** button. A meter window will appear, showing the progress of the operations.

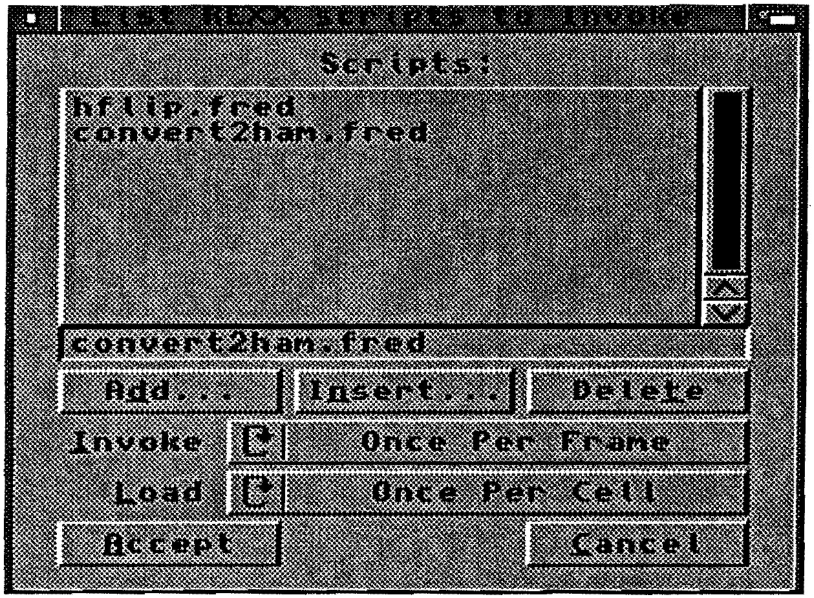


Figure 10.4: The Invoke ADPro control panel.

If you want to return to the FRED control screen without performing the operations, select **Cancel**.

Chapter 11

AnimOps

The **AnimOps** menu is accessible only when the backdrop is selected and contains the name of each program found in the AnimOps directory (usually `ADP_FRED:AnimOps`).

AnimOps are programs which read sequence files and generate ARexx commands automatically controlling MorphPlus (or ADPro). Any programmer may create an AnimOp. Contact ASDG for more details.

The AnimOps supplied by ASDG with FRED are described in their own sections.

In this chapter, we will discuss the AnimOps supplied by ASDG with FRED. While all AnimOps should have certain elements in common (that is, if their authors have followed ASDG's design guide), details concerning AnimOps supplied by third parties will not be contained here.

11.1 Compositor

The Compositor allows you to easily create combinations of fades, wipes, and composites between two or more sequences of images using FRED. Users of previous version of FRED may be familiar with the Compositor and Alpha_Compositor AnimOps that were included with the first versions of the program. This Compositor superseeds both of them, combining them into one AnimOp and including more options and controls.

The Compositor allows you to work on compositing "projects," which can be created, stored, modified, and retrieved. Select the **Compositor** menu item

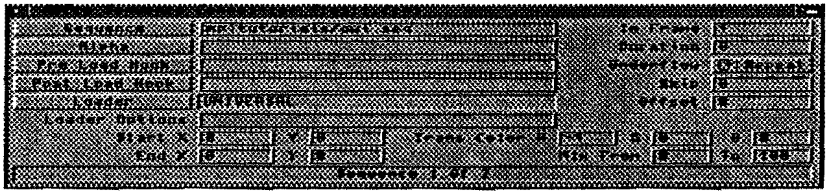


Figure 11.1: The Compositor AnimOp's Components control panel.

from the **AnimOps** menu. A thin window bar will appear atop the FRED screen. All of the main controls are available from the AnimOp's menu bar.

11.1.1 Projects

To create a new compositing project, select the **New...** menu item. The small AnimOp window bar will expand to the Results control panel. This panel will be discussed in Section 11.1.3.

Projects can be created, stored on disk, opened, closed, and edited. All of the controls for projects are available in the **Project** menu.

To define the compositing of sequences, you must go through two steps:

1. specify the sequences to be composited
2. specify the resulting sequence to create

The first step will be described in Section 11.1.2, while the second will be in Section 11.1.3.

11.1.2 Components

To display the Components control panel (as shown in Figure 11.1), select the **Define Components** menu subitem (from the **Mode** menu item). It is in this panel where you specify how one or more sequences will be composited together.

You have complete control over when, where, and by how much each sequence's frames will mix with the frames of other sequences. Let's take a look at the contents of this panel.

The first thing you should spot is the **Sequence Indicator** text field at the bottom of the panel. It is here where you can identify one sequence from another. You can flip from sequence definition to definition by using the **Next** and **Previous** menu items (in the **Edit** menu). This field will display which sequence (of the total number of sequences) you are currently editing.

Sequences are numbered starting from 1. The sequence defined as Sequence 1 can be thought of as being at the bottom (position-wise) of the sequences that will be defined. The other menu items in the **Edit** menu allow you to add, delete, copy, paste, and erase sequences from this "stack" of sequences.

The sequence file is described in the **Sequence** input field. You can either enter a new filename into the field or press the button to select one from the file requester. Only files ending in `.seq` will be visible. When saving sequence files out of FRED or any other program which outputs sequence files, you must ensure that this filename extension exists for FRED and these other programs to recognize it as a valid sequence file.

You can specify the transparency of the images in the sequence with another sequence of grayscale images (of the same dimension as the images in the sequence to composite) by entering the sequence's filename in the **Alpha** input field. This alpha channel sequence file should be of the same length as the primary sequence itself.

The **Pre Load Hook** and **Post Load Hook** input fields can contain the filenames of ARexx scripts which will be executed before and after loading each frame of the sequence. This feature gives you a lot of flexibility in manipulating the images in your sequence, without having to modify the originals—you can use the same master sequence with different hooks to produce a potentially drastic difference set of images.

The **Loader** and **Loader Options** text and input fields allow you to specify the type of images that are contained within the current sequence, and any loader-specific arguments which might be required. By default, the UNIVERSAL loader is used.

The **Start X** and **Y** values control the initial position of the upper left hand corner of the images in this sequence, relative to the currently defined "backdrop" image. The **End X** and **Y** values specify the final position of this same corner for the last frame to be processed in this sequence.

The "backdrop" image for a given frame number will not always be the image defined in the bottommost sequence (Sequence 1). Depending upon the **In Frame** number (described later in this section), a "higher" sequence might actually constitute the backdrop.

The transparent color in the current sequence's images can be defined with the **Trans Color R**, **G**, and **B** values. If **R** is set to -1, then there will be no

transparent color (the image will be opaque). If each of these values are set to a number between 0 and 255, then any pixel in the image that is this color will be transparent, and that the next lowest opaque pixel (from lower sequences) will "show through." Note that if a current frame is the backdrop image, then these transparency values are ignored.

The amount of mixing that will be done, from the first frame to process to the last, is defined in the **Mix From** and **To** input fields. These values can range between 0 and 100.

The actual frames (in this sequence) that will be composited is controlled by the values in the input fields and cycle gadget located at the right of the panel.

In Frame describes at which frame number (sequences are described as starting with a frame number of 1) in the resulting sequence this current sequence's frames will start to be composited. The first frame (in this sequence) to process is specified in the **Offset** input field. The number of frames to process is specified in the **Duration** field. If you want to process every other frame, or every third or fourth frame, then you can change the value in the **Skip** input field to tell the AnimOp how many frames to skip after finishing with a particular frame.

If the resulting sequence "fills up with images" (the total number of frames requested has already been reached) but you still have frames in the current sequence to process, then those frames will not affect the final sequence at all. If, however, you use up all of the selected frames in the sequence before the resulting sequence finishes, then the setting of the **Underflow** cycle gadget will define what to do with the current sequence.

If set to **Repeat**, then the last frame that was processed will be used for the rest of the resulting sequence. If set to **Loop**, then the current sequence will act as if the sequence was never-ending (i.e., the sequence would start at the beginning again). If set to **Stop**, then no more frames from this sequence will take part in the compositing.

11.1.3 Results

Once you have defined the sequences that will be composited together, you would then define the resulting sequence that will be generated. Switch to the Results mode by selecting the **Define Results** menu subitem. The Compositor's Results control panel (as shown in Figure 11.2) will appear.

The name of the resulting sequence must be defined in the **Output Sequence** input field. The composited frames will be saved (appended by the sequence's current frame number) with a base (root) name, as defined in the **Image Root Name** input field. If you want the frames to be saved as

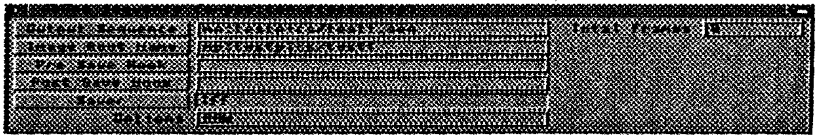


Figure 11.2: The Compositor AnimOp's Results control panel.

`Work:Images/vacation.00001`, `Work:Images/vacation.00002`, and so forth, then enter `Work:Images/vacation` as the root name.

If you require special processing of the image before or after it is saved to disk, then specify the ARexx scripts in the **Pre Save Hook** and **Post Save Hook** fields, respectively.

The format in which these images will be saved must be specified in the **Saver** text field. Any saver-specific options that need to be included should be entered into the **Saver Options** input field.

Lastly, the total number of frames in the resulting sequence is defined by the value in the **Total Frames** input field.

Once you are satisfied with the project, you may want to save it before actually initiating the compositing. To begin the process, select the **Accept** menu item (from the **Project** menu). To exit this AnimOp, select **Quit**.

Chapter 12

MorphPlus Reference

This chapter describes all the control panels, menu items, and ARexx commands available in MorphPlus.

12.1 Starting MorphPlus

The MorphPlus program can be invoked by either double-clicking on its Workbench icon or typing its name at a Shell prompt. You can also customize various parts of the program by specifying some Tool Types or using a few command line arguments.

■ Tool Type Options

You can specify Tool Type options (using MorphPlus' icon) to control MorphPlus. These include:

`BEHIND=`*n* or `BH=`*n*
: If *n* is non-zero, the MorphPlus main screen will be opened behind all other screens open at the time.

`MAXMEM=`*n* or `MM=`*n*
: If present, MorphPlus will be limited to using *n* bytes for its image buffer size. Do not include comma separators in the *n* value. If there is less than *n* bytes free, the buffer size will be equal to the available number, not *n*.

`NOENHANCED=`*n* or `NE=`*n*
: If *n* is non-zero, the enhanced palette option will be disabled.

DEFAULTFILE=*filename* or **DF=***filename*

If present, the defaults will be loaded from the *filename* specified.

NOLOADDEFAULTS=*n* or **NL=***n*

If *n* is non-zero, the **ADProDefaults** file will not be loaded when MorphPlus is started.

NOSAVEDEFAULTS=*n* or **NS=***n*

If *n* is non-zero, the **ADProDefaults** file will not be saved when MorphPlus exists.

Command Line (Shell) Arguments

Alternatively, you can use one of more of the Command Line (Shell) options :

BEHIND

If present, the MorphPlus screen will be opened behind all other screens.

MAXMEM=*n* or **MM=***n*

If present, MorphPlus will be limited to using *n* bytes for its image buffer size. Do not include comma separators in the *n* value. If there is less than *n* bytes free, the buffer size will be equal to the available number, not *n*.

NOENHANCED or **NE**

If present, the enhanced palette option will be disabled.

DEFAULTFILE=*filename*

If present, the defaults will be loaded from the *filename* specified.

NOLOADDEFAULTS

If present, the **ADProDefaults** file will not be loaded when MorphPlus is started.

NOSAVEDEFAULTS=*n*

If present, the **ADProDefaults** file will not be saved when MorphPlus exits.

12.2 MorphPlus Control Screen

MorphPlus' main (control) screen contains all of the controls for loading and saving images and selecting and executing operators. The screen and colors controls, as well as various other miscellaneous controls are available from the main screen.

12.2.1 Commands

ARexx Interface

Control over the Load and Save Formats, the Operator Type, as well as the Load, Save, and Execute Op operations will be discussed in their own sections.

Render Type

Syntax: `RENDER_TYPE` (1)
`RENDER_TYPE` *rvalue* (2)

The first form of the command returns the current rendering type. This result will be returned in `ADPRO_RESULT` and may be one of the following: `2`, `4`, `8`, `16`, `32`, `64`, `128`, `256`, `EHB`, `HAM`, `HAM8` or `CUST`.

The second form of the command allows the rendering type to be set to the supplied string which must match one of the above mentioned values.

If string was	RC	ADPRO_RESULT
accepted	0	the *previous* rendering type
not accepted	10	"Invalid Argument To Function"

For more information about rendering type, please see Section 4.4.5.

Execute

Syntax: `EXECUTE`

This command is the analog of the **Execute** button on the MorphPlus main screen. It causes rendered data to be created from raw data.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

For more information about the **Execute** button, please refer to Section 4.2.2.

Displaying a Rendered Image

Syntax: `ADPRO_DISPLAY`

This command is equivalent to the **ReDisplay** button on the MorphPlus main screen. Executing this command will cause any Amiga displayable rendered image to pop to front. Images rendered in the A-RES modes cannot be displayed using this command.

If	RC	ADPRO_RESULT
an image was displayed	0	undefined
only A-RES mode Amiga displayable data is available	10	"Error"

Ending the Display of a Rendered Image

Syntax: `ADPRO_UNDISPLAY`

This command will move the screen containing any Amiga displayable image, which might be currently defined, to the backmost screen. The screen which will be revealed may not necessarily be the MorphPlus main screen. If this is required, you must issue an `ADPRO_TO_FRONT` as well.

Operation	RC	ADPRO_RESULT
always returns	0	undefined

MorphPlus Version

Syntax: `VERSION`

This command returns the version identification string of the currently executing MorphPlus.

Operation	RC	ADPRO_RESULT
always returns	0	version string

Serial Number

Syntax: `SERIAL_NUMBER`

This command returns the serial number of this copy of MorphPlus in `ADPRO_RESULT`. The serial number can be displayed by clicking on the **About** button on the main screen. This command will never return an error.

Operation	RC	ADPRO_RESULT
always returns	0	serial number

Determining Image Memory Size

Syntax: **CURRENT_MEM_SIZE**

This command will return the number of bytes of main memory in use by MorphPlus' primary image memory buffer. This value is returned in **ADPRO_RESULT**.

This value is reported from the user interface in the **About** panel.

Operation	RC	ADPRO_RESULT
always returns	0	image buffer size

Exiting MorphPlus

Syntax: **ADPRO_EXIT**

This command causes MorphPlus to terminate. Clearly, any ARexx command destined for MorphPlus executed after this command will not work, since MorphPlus will no longer be present to service it.

Operation	RC	ADPRO_RESULT
always returns	0	unchanged

Bringing the MorphPlus Screen to Front

Syntax: **ADPRO_TO_FRONT**

This command causes the MorphPlus main control screen to pop in front of all other screens.

Operation	RC	ADPRO_RESULT
always returns	0	undefined

Pushing the MorphPlus Screen Behind

Syntax: `ADPRO_TO_BACK`

This command causes the MorphPlus main control screen to be pushed behind all other screens.

Operation	RC	ADPRO_RESULT
always returns	0	undefined

12.2.2 Balancing Controls

ARexx Interface

These commands interrogate or set values affecting color balancing.

Red Adjustment

Syntax: `RED` (1)
`RED` *value* (2)

The first form returns the current red adjustment setting.

The second format sets the red adjustment to the supplied value, which must be in the range of -50 to 50.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* red adjustment setting
out of range	10	"Invalid Argument To Function"

Note that if the value you wish to specify is a negative number, you must surround the negative number with quotation marks.

See Section 4.3.1 for more information about the red adjustment.

Green Adjustment

Syntax: `GREEN` (1)
`GREEN` *value* (2)

The first form returns the current green adjustment setting.

The second format sets the green adjustment to the supplied value, which must be in the range of -50 to 50.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* green adjustment setting
out of range	10	"Invalid Argument To Function"

Note that if the value you wish to specify is a negative number, you must surround the negative number with quotation marks.

See Section 4.3.1 for more information about the green adjustment.

Blue Adjustment

Syntax: `BLUE` (1)
`BLUE` *value* (2)

The first form returns the current blue adjustment setting.

The second format sets the blue adjustment to the supplied value, which must be in the range of -50 to 50.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* blue adjustment setting
out of range	10	"Invalid Argument To Function"

Note that if the value you wish to specify is a negative number, you must surround the negative number with quotation marks.

See Section 4.3.1 for more information about the blue adjustment.

Brightness Adjustment

Syntax: `BRIGHTNESS` (1)
`BRIGHTNESS` *value* (2)

The first form returns the current brightness adjustment setting.

The second format sets the brightness adjustment to the supplied value, which must be in the range of - 50 to 50.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* brightness adjustment setting
out of range	10	"Invalid Argument To Function"

Note that if the value you wish to specify is a negative number, you must surround the negative number with quotation marks.

See Section 4.3.1 for more information about the brightness adjustment.

Contrast Adjustment

Syntax: `CONTRAST` (1)
`CONTRAST` *value* (2)

The first form returns the current contrast adjustment setting.

The second format sets the contrast adjustment to the supplied value, which must be in the range of -50 to 50.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* contrast adjustment setting
out of range	10	"Invalid Argument To Function"

Note that if the value you wish to specify is a negative number, you must surround the negative number with quotation marks.

See Section 4.3.1 for more information about the contrast adjustment.

Gamma Adjustment

Syntax: `GAMMA` (1)
`GAMMA` *value* (2)

The first form returns the current gamma adjustment setting.

The second format sets the gamma adjustment to the supplied value, which must be in the range of 0 to 100.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* gamma adjustment setting
out of range	10	"Invalid Argument To Function"

NOTE: Although you can set a gamma value in the range of -50 to 50 in the user interface (the **Gamma** *slider and input field), the ARexx interface must be specified in the range of 0 to 100, where 50 means no gamma correction.*

See Section 4.3.1 for more information about the gamma adjustment.

12.2.3 Palette Controls

■ **ARexx Interface** The commands in this subsection refer to and manipulate the settings in the Palette control panel.

Palette Width

Syntax: `PWIDTH` (1)
`PWIDTH (NORMAL | ENHANCED)` (2)

The first form of the command returns the current width (accuracy) of the palette in the `ADPRO_RESULT` variable.

The second form of the command allows the ARexx programmer to set the palette width to the supplied value, which must be either `NORMAL` or `ENHANCED`.

If value was	RC	ADPRO_RESULT
either `NORMAL` or `ENHANCED`	0	the *previous* palette depth
neither `NORMAL` nor `ENHANCED`	10	"Invalid Argument To Function"

Note then when the palette is locked and you switch from `ENHANCED` to `NORMAL`, some resolution is lost. Switching back does not regain this lost resolution. When the palette is unlocked, palette recomputation will be done at the current palette width.

For more information concerning palette width, please refer to Section 4.3.2.

Palette Status

Syntax: `PSTATUS` (1)
`PSTATUS (LOCKED | UNLOCKED)` (2)

The first form returns the string LOCKED or UNLOCKED in the ADPRO_RESULT variable, provided that results have been requested.

The second form sets the status of the palette to the specified value, which must be the string LOCKED or UNLOCKED.

If value was	RC	ADPRO_RESULT
either LOCKED or UNLOCKED	0	the *previous* state of the palette
neither LOCKED nor UNLOCKED	10	"Invalid Argument To Function"

For more information about palette status, refer to Section 4.3.2.

Total Number of Colors

Syntax: PTOTAL (1)
PTOTAL *value* (2)

The first form returns the currently defined total number of colors in ADPRO_RESULT provided that results have been requested.

The second form sets the total number of colors to the supplied value which must be one of the following: 2, 4, 8, 16, 32, 64, 128, 256, EHB, HAM, or HAM8.

When specifying HAM, the actual total number of colors possible refers to the number of color registers available in the standard HAM mode (which is 16). Similarly, when referring to an EHB screen, the actual total number of colors available is 32.

If value was	RC	ADPRO_RESULT
accepted	0	total number of colors
out of range	10	"Invalid Argument To Function"

Note that the total number of colors, the offset of color zero, and the number of colors to be used must be consistent. For example, an error will be generated if you request a total number of colors which is smaller than the currently defined number of colors used.

For more information about the total number of colors, please see Section 4.3.2.

Number of Colors Used

Syntax: `PUSED` (1)
`PUSED` *value* (2)

The first form returns the current number of colors to be used in `ADPRO_RESULT` provided that results have been requested.

The second form sets the number of colors to be used to the supplied value which must range between 2 and the total number of colors.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* number of colors used
out of range	10	"Invalid Argument To Function"

Note that the total number of colors, the offset of color zero, and the number of colors to be used must be consistent. For example, an error will be generated if you request a number of colors to be used which is larger than the total number of colors.

For more information about the number of colors to be used, please see Section 4.3.2.

Offset Color Zero

Syntax: `POFFSET` (1)
`POFFSET` *value* (2)

The first form returns the currently defined offset of color zero in `ADPRO_RESULT` provided that results have been requested.

The second form sets the offset of color zero to the supplied value, which must range from 0 to the total number of colors minus one. When in `HAM` mode, the maximum value for the offset of color zero is 14. When in `HAM8` mode, the maximum value for the offset of color zero is 62. When in `EHB` mode, the maximum is 30.

If value was	RC	ADPRO_RESULT
accepted	0	offset value
out of range	10	"Invalid Argument To Function"

Note that the total number of colors, the offset of color zero, and the number of colors to be used must be consistent. For example, an error will be generated if you request an offset of color zero which would overflow the total number of colors presently defined when the number of colors to be used is factored in.

For more information about the offset of color zero, please refer to Section 4.3.2.

Palette Contrast

Syntax: `PCONTRAST` (1)
`PCONTRAST (0 | 1)` (2)

The first form of the command returns the present setting of the palette contrast button. If `ADPRO_RESULT` contains a 0, then the palette contrast button is set to **No Cont**. Otherwise, the palette contrast button is set to **Cont**.

The second form allows you to set the state of the palette contrast button. If you supply an argument of 0, then the palette contrast button is set to the off or **No Cont** setting. If you supply an argument of 1, then the palette contrast button is set to the on or **Cont** setting.

If value was	RC	ADPRO_RESULT
either 0 or 1	0	undefined
neither 0 nor 1	10	undefined

For more information about the palette button, please refer to Section 4.3.2.

Palette Sort Order

Syntax: `PSORT` (1)
`PSORT (0 | 1)` (2)

The first form of the command returns the present palette sort order. If `ADPRO_RESULT` contains a 0, then the sort order is ascending. Otherwise, the sort order is descending.

The second form sets the palette sort order to ascending or descending depending upon whether the supplied argument is a 0 or 1, respectively.

If value was	RC	ADPRO_RESULT
either 0 or 1	0	the *previous* sort direction
neither 0 nor 1	10	"Invalid Argument To Function"

For more information concerning palette sorting, please refer to Section 4.3.2.

Setting/Getting Palette Contents

Syntax: `PPOKE` *register* (1)
`PPOKE` *register red green blue* (2)

The first form of the command returns a string describing the contents of the specified color register. The string, returned in `ADPRO_RESULT`, is formatted as the red, green, and blue values of the color register separated by spaces and provided in decimal in the range of 0 to 255.

The second form of the command allows you to set the contents of the specified color register with the supplied values. Range checking will be performed on the supplied color values, which should range from 0 to 255.

For both forms, range checking will be performed on the supplied register number.

If operation was	RC	ADPRO_RESULT
successful	0	"*red green blue*"
not successful	non-zero	string indicating the nature of the error

While the allowable range of color values is from 0 to 255 (24 bit accuracy), MorphPlus' internal storage of the colors you define may vary according to whether you are in a normal or enhanced palette mode. As a consequence, it is normal to read a slightly different value than what might have been written immediately before.

You can convert 12 bit colors (the ones you would use with Deluxe Paint, for example) into 24 bit colors by multiplying each color by 17. So, for example, 15 becomes 255 while 1 would become 17.

Getting the Workbench Palette

Syntax: `PGETWB`

This command attempts to load the currently defined Workbench palette.

Note that this command results in the loading of up to four colors only. Should you require access to a deeper Workbench palette (possible under Workbench 2.0 and later), you can use the `PLOAD` command instead.

The Workbench palette is loaded starting with the color register specified by `POFFSET`.

Operation	RC	ADPRO_RESULT
always returns	0	unchanged

For more information about this command, please refer to Section 4.3.2.

Loading a Palette

Syntax: **PLOAD** (1)
PLOAD *filename* (2)

If no filename is supplied (as in the first form), then MorphPlus will bring up its file requester so that the user can manually enter the name of a palette file to be loaded. Note that MorphPlus will automatically pop its screen to front whenever it displays its file requester. Therefore, it is the ARexx programmer's responsibility to depth-arrange screens after calling this function with no supplied filename.

The second form of this command attempts to load a palette from the named file. The supplied file must be in IFF format but does not have to contain any image data.

If palette was	RC	ADPRO_RESULT
loaded	0	empty string
not loaded	10	empty string

Note that loading a palette is directly affected by the total number of colors, the number of colors used, and the offset of color zero.

For more information concerning the loading of a palette, please see Section 4.3.2.

Saving a Palette

Syntax: **PSAVE** (1)
PSAVE *filename* (2)

If a filename is not supplied (as in the first form), MorphPlus will display its file requester to allow the user to input a filename manually. Remember to handle the case of the user not making any file selection at all.

The second form of this command attempts to save the currently defined palette using the provided filename. The resulting file will be encoded in the IFF format.

If palette was	RC	ADPRO_RESULT
saved	0	empty string
not saved	10	empty string

It is the ARexx programmer's responsibility to prevent the unwanted overwriting of a pre-existing file.

For more information about saving a palette, please refer to Section 4.3.2.

The Wrong Way to Control the Palette

In each of the definitions for PTOTAL, PUSED, and POFFSET, a warning is given that the total number of colors, the number of colors to be used, and the offset of color zero must be consistent.

For example, the following is one of many wrong ways to set the total number of colors to 16 with 5 used and a color zero offset of 3:

```
POFFSET 3
PTOTAL 16
POFFSET 3
PUSED 5
```

The above example will fail if the number of colors to be used had an initial value above 12. This is because setting the offset of color zero to 3 would exceed the total number of colors if the number of colors to be used was 13 or higher.

Pathological cases abound, where the order in which these values are set will determine if an error occurs or not. Setting the offset of color zero to zero (then setting the other desired values) always increases your safety.

Make use of the query versions of the palette commands prior to setting palette values to avoid running afoul of MorphPlus' internal consistency check ($PUSED + POFFSET \leq PTOTAL$).

12.2.4 Screen Controls

ARexx Interface

Screen Type

Syntax: `SCREEN_TYPE` (1)
`SCREEN_TYPE` *mask* (2)

The first form of the command returns the current screen type in ADPRO_RESULT. The value returned is in the form of a bit-mask comprised of the values indicated in Table 12.1.

The second form of the command allows the screen type to be set to the supplied value. The supplied value must be within the range of 0 to 31.

Mode	Value
Hi Resolution On	1
Interlace On	2
PAL On	4
Horizontal Overscan On	8
Vertical Overscan On	16
VGA Mode On	32
Super Hi-Res Mode On	64

Table 12.1: Mask values for setting the screen type.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* screen type
out of range	10	"Invalid Argument To Function"

Your ARexx program might include the following:

```
VOVERSCAN = 16
HIRES = 1
INTERLACE = 2
PAL = 4
HOVERSCAN = 8
VOVERSCAN = 16
```

Then, elsewhere in your ARexx program you can say things like:

```
SCREEN_TYPE HIRES + INTERLACE + HOVERSCAN + VOVERSCAN
```

This command would set the screen type to high resolution, NTSC, interlace, with both horizontal and vertical overscan.

Note that some modes cannot be used with some others. Specifically the flags for VGA and PAL are mutually exclusive and the flags for Super Hi-Res and Hi-Res are mutually exclusive. If you specify these combinations, an error will result, which sets `ADPRO_RESULT` to `Invalid Argument To Function`.

Another possible error condition is setting a legal combination of flags which simply aren't supported on the Amiga you are currently using. For example, asking for Super Hi-Res on a machine without Super Hi-Res capability will fail. The error returned in `ADPRO_RESULT`, which indicates that the specified flags were legal but that MorphPlus could not comply, is simply `ERROR`.

ID	Method
0	None
1	Floyd-Steinberg
2	Burkes
3	Sierra
4	Jarvis
5	Stucki
6	Random

Table 12.2: Dither methods and their identifiers.

Dither Selection

Syntax: `DITHER` (1)
`DITHER` *method_number* (2)

The first form returns the identifier of the currently selected dither method. The dither methods currently defined are given in Table 12.2.

The second form sets the dither method to the method corresponding to the supplied value, which must be in the range of 0 to 6.

If value was	RC	ADPRO_RESULT
accepted	0	the *previous* dither method
out of range	10	"Invalid Argument To Function"

See Section 4.4.4 for more information about dithering.

12.2.5 Other ARexx Commands

This section describes commands which are available through ARexx which do not have direct equivalents in the MorphPlus user interface or don't fall into any of the previously described categories.

Last Saved Image

Syntax: `LAST_SAVED_IMAGE`

This command returns (in `ADPRO_RESULT`) the entire path name used during the last manually executed **Save** command. A manually executed **Save** is one *not* performed from ARexx.

If manually performed **Save** or **Load**	RC	ADPRO_RESULT
has been performed	0	path name
has not been performed	10	undefined

Note that if the user has not performed a **Save** by the time the first **Load** is executed, MorphPlus will copy the **Load** filename to the **Save** filename. This is the only circumstance in which LAST_SAVED_IMAGE will return a filename which wasn't actually saved.

Last Loaded Image

Syntax: LAST_LOADED_IMAGE

This command returns (in ADPRO_RESULT) the entire path name used during the last manually executed **Load** command. A manually executed **Load** is one *not* performed from ARexx.

If manually performed **Load** or **Save**	RC	ADPRO_RESULT
has been performed	0	path name
has not been performed	10	undefined

Note that if the user has not performed a **Load** by the time the first **Save** is executed, MorphPlus will copy the **Save** filename to the **Load** filename. This is the only circumstance in which LAST_LOADED_IMAGE will return a filename which wasn't actually loaded.

Image Width

Syntax: XSIZE

This command returns the width of the currently loaded image in ADPRO_RESULT.

Operation	RC	ADPRO_RESULT
always returns	0	width of image

Image Height

Syntax: YSIZE

This command returns the height of the currently loaded image in `ADPRO_RESULT`.

Operation	RC	ADPRO_RESULT
always returns	0	height of image

Determining Currently Available Data

Syntax: `IMAGE_TYPE`

This command returns a string describing what kinds of data are available within MorphPlus at the present time. The string is returned in `ADPRO_RESULT` and can have the following forms:

NONE

This means that no image data (either raw or rendered) is currently available within MorphPlus.

COLOR

This means that raw color data is available but no rendered data is available at the current time.

GRAY

This means that raw grayscale data is available but no rendered data is available at the current time.

BITPLANE

This means that rendered image data is available but that no raw color or grayscale data is presently available.

COLOR BITPLANE

This means that both raw color and rendered image data are currently available.

GRAY BITPLANE

This means that both raw grayscale and rendered image data are currently available within MorphPlus.

Operation	RC	ADPRO_RESULT
always returns	0	NONE, COLOR, GRAY, BITPLANE, COLOR BITPLANE, or GRAY BITPLANE

The ARexx programmer must not make any assumptions about the order in which the words in the result string will appear. Instead of direct string comparisons to determine the contents of the result string, use ARexx's substring functions.

Pausing a Specified Time

Syntax: **PAUSE** *ticks*

This command "kills time" for the specified number of ticks. A tick lasts for one-fiftieth of a second. The specified value must be in the range of 1 to 180000.

If value was	RC	ADPRO_RESULT
accepted	0	unchanged
out of range	10	"Invalid Argument To Function"

Getting a String From the User

Syntax: **GETSTRING** (1)
GETSTRING *title* (2)
GETSTRING *title default* (3)

This command can be used to query the user for a string to be returned into your ARexx program.

The first form of the command displays a string requester with a default title and no default text.

The second form uses the supplied string for the title bar of the string requester.

The third form uses the supplied strings for the title bar of the string requester and the default contents of the string requester, respectively.

If a string was	RC	ADPRO_RESULT
entered	0	the entered string
not entered	10	undefined

It is the ARexx programmer's responsibility to move MorphPlus to front (using **ADPRO_TO_FRONT**) before calling this function.

Note that strings with spaces must be quoted properly as in the following example:

```
GETSTRING 'Spaces' '"Require Double Quoting!"'
```

Getting a Number From the User

Syntax: **GETNUMBER** (1)
GETNUMBER *title* (2)
GETNUMBER *title default* (3)
GETNUMBER *title default min max* (4)

This command can be used to get a signed integer from the user.

The first form of the command displays an integer requester with a default title and no default value.

The second form uses the supplied title and no default value.

The third form uses the supplied title and the supplied default value.

The fourth form allows you to specify a minimum and maximum value between which the users input must fall.

If a number was	RC	ADPRO_RESULT
entered	0	the entered value
not entered	10	undefined

It is the ARexx programmer's responsibility to move MorphPlus to front (using `ADPRO_TO_FRONT`) before calling this function.

Note that strings with spaces must be quoted properly as in the following example:

```
GETNUMBER '"Spaces Require Double Quoting!"'
```

Getting a Filename From the User

Syntax: **GETFILE** (1)
GETFILE *title* (2)
GETFILE *title default_dir* (3)
GETFILE *title default_dir default_file* (4)

This command can be used to get a filename from the user.

The first form of the command displays the file requester with a default title and no default filename.

The second form uses the supplied title and no default filename.

The third form uses the supplied title and the supplied default directory name. That is, the file requester will load the contents of the specified default directory.

The fourth form allows you to specify a title, a default directory, as well as a default filename.

If a file was	RC	ADPRO_RESULT
selected	0	the selected filename
not selected	10	undefined

It is the ARexx programmer's responsibility to move MorphPlus to front (using `ADPRO_TO_FRONT`) before calling this function.

Note that strings with spaces must be quoted properly as in the following example:

```
GETFILE '"Spaces Require Double Quoting!"'
```

Getting a List of Filenames From the User

Syntax: `GETFILES` (1)
`GETFILES` *title* (2)
`GETFILES` *title default_dir* (3)
`GETFILES` *title default_dir default_file* (4)

This command can be used to get a list of filenames from the user.

The first form of the command displays the file requester with a default title and no default filename.

The second form uses the supplied title and no default filename.

The third form uses the supplied title and the supplied default directory name. That is, the file requester will load the contents of the specified default directory.

The fourth form allows you to specify a title, a default directory, as well as a default filename.

If	RC	ADPRO_RESULT
at least one file was selected	0	the selected filenames, separated by quotation marks
no file was selected	10	undefined

To select more than one file using the file requester, simply hold down either `Shift` key while clicking on filenames. All filenames selected must be from the same directory. This command makes it easier to write ARexx programs which will batch process multiple files, since it gives you a way of allowing the user to specify multiple files at one time.

It is the ARexx programmer's responsibility to move MorphPlus to front (using `ADPRO_TO_FRONT`) before calling this function.

Note that strings with spaces must be quoted properly as in the following example:

```
GETFILES '"Spaces Require Double Quoting!"'
```

Following, is an examples of using `GETFILES` and parsing its results.

```
/*
** An example of using GETFILES.
*/

OPTIONS RESULTS

ADDRESS "ADPro"

GETFILES
IF RC ~= 0 THEN DO
   /* No selection */
   EXIT
END

LIST = ADPRO_RESULT
COUNT = WORDS( ADPRO_RESULT )
I = 1

DO WHILE (I <= COUNT)
   OKAY1 WORD( LIST, I )
   I = I + 1
END
```

Getting a Directory Name From the User

Syntax:	`GETDIR`	(1)
	`GETDIR` *title*	(2)
	`GETDIR` *title default*	(3)

This command can be used to get a directory name or path from the user. This is especially handy when you wish to specify a directory in which your input (or output) frames may be found (or placed).

The first form of the command displays the file requester with a default title and no default filename.

The second form uses the supplied title and no default filename.

The third form uses the supplied title and the supplied default filename.

If a directory was	RC	ADPRO_RESULT
selected	0	the selected path
not selected	10	undefined

This command uses the same file requester as used elsewhere in this program. However, only directories will be shown to the user. Ordinary files will be ignored.

It is the ARexx programmer's responsibility to move MorphPlus to front (using `ADPRO_TO_FRONT`) before calling this function.

Note that strings with spaces must be quoted properly as in the following example:

```
GETDIR '"Spaces Require Double Quoting!"'
```

The OKAY1 Requester

Syntax: `OKAY1` "*string*"

The `OKAY1` command can be used to display an arbitrary string. The string will be rendered on the MorphPlus screen and execution will pause until the user selects the "resume" button in the `OKAY1` requester.

Operation	RC	ADPRO_RESULT
always returns	1	undefined

It is the ARexx programmer's responsibility to move MorphPlus to front (using `ADPRO_TO_FRONT`) before calling this function.

The OKAY2 Requester

Syntax: `OKAY2 "string"`

The **OKAY2** command can be used to display an arbitrary string and solicit a two valued answer from the user. The string will be rendered on the MorphPlus screen and execution will pause until the user selects the "OK" or "Cancel" buttons in the **OKAY2** requester.

If the user selects	RC	ADPRO_RESULT
"Cancel"	0	undefined
"OK"	non-zero	undefined

It is the ARexx programmer's responsibility to move MorphPlus to front (using **ADPRO_TO_FRONT**) before calling this function.

Pixel Aspect

Syntax: `PIXEL_ASPECT`

This command returns the current pixel aspect ratio in **ADPRO_RESULT**.

If an image was	RC	ADPRO_RESULT
available	0	pixel aspect
not available	10	unchanged

Pixel Resolution

Syntax: `PIXEL_RESOLUTION`

This command returns the current pixel resolution (horizontal and vertical DPI) in **ADPRO_RESULT**.

If an image was	RC	ADPRO_RESULT
available	0	pixel resolution
not available	10	unchanged

Obtaining Exclusive Use of MorphPlus

Syntax: `OBTAIN_ADPRO`

This command tries to place an exclusive "lock" on MorphPlus, so that your script will have sole access of the program. This is very important when two or more scripts are likely to call MorphPlus at the same time.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

Releasing Exclusive Use of MorphPlus

Syntax: `RELEASE_ADPRO`

This command will release the exclusive "lock" on MorphPlus, so that other scripts may be able to lock the program.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

Starting MorphPlus Via ARexx

The following ARexx fragment can be adapted for use in your own ARexx programs. This code attempts to locate a currently running MorphPlus. Failing that, it will attempt to start up MorphPlus. In order to do this, `ADPRO:` must be defined.

```
** $VER: Locate_MorphPlus.rexx 1.00 (1.9.92)
**
** This ARexx program will attempt to find
** a currently running MorphPlus. If one is
** not found, then it will attempt to start
** up MorphPlus.
**
** The main guts of this program are
** imbedded in a sub-routine to make it
** easier to glue into your own code.
**
** Example ARexx program for controlling
** MorphPlus by ASDG, Incorporated.
```

```
** Copyright 1992 ASDG, Incorporated.
** All Rights Reserved Worldwide
*/

OPTIONS RESULTS

CALL Locate_MorphPlus

IF RESULT = 1 THEN
   SAY "MorphPlus has been found"
ELSE
   SAY "Could not locate or start MorphPlus"
EXIT


Locate_MorphPlus:

Max_Seconds_To_Load = 60
Flag = 0
LibName = 'rexxsupport.library'

IF POS(LibName, SHOW('Libraries')) = 0 THEN
   ADDLIB(LibName, 0, -30, 0)
IF POS(LibName, SHOW('Libraries')) = 0 THEN
   RETURN 0

IF STATEF('ADPRO:') = "" THEN
   RETURN 0

TIME('R')
DO WHILE ((TIME('E') < Max_Seconds_To_Load) &
          (POS('ADPro', SHOW('Ports')) = 0))
   IF Flag = 0 THEN DO
      ADDRESS COMMAND
         'run < nil: > nil: ADPRO:MorphPlus'
      Flag = 1
   END
   ADDRESS COMMAND 'WAIT 1'
END
IF POS('ADPro', SHOW('Ports')) = 0 THEN
   RETURN 0
ELSE
   RETURN 1
```

12.3 Loaders

Loaders can be selected from the main screen or through their ARexx interfaces. For specific information, read each loader's control panel and ARexx interface sections.

12.3.1 General Information

ARexx Interface

Load Format

Syntax: `LFORMAT` *format*

The first form of the command will fill the `ADPRO_RESULT` field with the name of the currently selected Load Format, if results are requested.

The second form of the command sets the Load Format to the specified string. MorphPlus will qualify the supplied string as being one of `"IFF"`, `"ALPHA"`, `"ANIM"`, `"RIPPLE"`, or `"TEMP"`.

If module	RC	ADPRO_RESULT
exists	0	the *previously* selected Load Format
does not exist	10	"Unknown Module"

Refer to Section 2.3.1 and Chapter 5 for more information concerning Load Formats.

Calling a Loader

Syntax: `LOAD` *filename* [*loader_opts*] [*comp_opts*] `[FORCE_PALETTE]`

This command causes the specified file to be loaded. The only required argument is the filename to be loaded. If you wish to allow the user to manually enter a filename to load, then you must supply a NULL *filename* (in ARexx, this would be: `'"' || '"'`).

The loader which will be invoked is specified with the `LFORMAT` command.

The *loader_opts* arguments are loader dependent. The currently defined loader-specific arguments are defined in each loader's ARexx Interface reference section.

comp_opts define the ARexx interface to the image compositing capability of MorphPlus. *comp_opts* may be specified in one of two formats (old and new).

Old Style

left_edge top_edge mix_level (1)
left_edge top_edge mix_level rt gt bt (2)

In both forms, you must specify where the image to be loaded will be placed relative to the upper left-hand corner of the image currently in memory. This is specified by *left_edge* and *top_edge*. Also, both forms require that you specify the mix level to be used (which may range in value from 1 to 100).

The second allows you to specify which 24 bit-plane color value will be considered transparent. If you are merging a grayscale image, the gray level to be considered transparent is taken to be *rt*. In the first form, transparency is disabled.

New Style

The new style compositing arguments can be placed anywhere on the argument line after the *filename* argument. Zero or more of the following can be defined:

`COMPOFFSET` *left top*

Specify the image offset of the foreground image relative to the background image.

GUI Equivalent:	**Off X** and **Off Y** values.
Constraints:	None
Required:	No

`COMPMIX` *mix*

Specify the mix percentage.

GUI Equivalent:	**Mix** value.
Constraints:	$0 \leq mix \leq 100$
Required:	No

`COMPTRANS` *r g b*

Specify the transparent color (in the foreground image).

GUI Equivalent:	**R**, **G**, and **B** values.
Constraints:	$-1 \leq r \leq 255$ $-1 \leq g \leq 255$ $-1 \leq b \leq 255$
Required:	No

`NOPAD`

Specify that raw data should not be created.

GUI Equivalent:	None
Constraints:	None
Required:	No

For more information about image compositing, please refer to Section 2.3.4.

The `FORCE_PALETTE` argument allows the loaded image's palette (if it has one) to replace the current palette even if the palette is currently locked. The `FORCE_PALETTE` option, if desired, must be the last option specified.

If the load operation is successful, `RC` = 0.

Some examples follow:

```
LOAD Filename
```

This simply loads the file specified by the variable `Filename`. This image completely replaces any previous image in memory.

```
LFORMAT "IFF"
LOAD Filename 0 0 100 255 255 255
```

This shows an example of loading an IFF-ILBM file with the image compositing options present. This will load the image at an offset of 0,0. The mix level is 100 percent and pure white will be considered to be transparent.

Load Orientation

Syntax: `ORIENTATION` (1)
`ORIENTATION (PORTRAIT | LANDSCAPE)` (2)

The first form of the command returns the current load orientation. The `ADPRO_RESULT` variable will be set to either `PORTRAIT` or `LANDSCAPE`.

The second and third forms of the command set the load orientation to the specified value.

For more information about the uses of portrait and landscape orientations, please refer to Section 2.3.2.

12.3.2 ALPHA

The ALPHA loader lets you use a grayscale IFF-ILBM image as the alpha channel when compositing an image over the current image in MorphPlus. It

can be accessed through the main control screen and through ARexx.

ARexx Interface

Syntax: `LFORMAT "ALPHA"`
`LOAD` *image_to_load* *loader_opts*

The only *loader_opts* for the ALPHA loader is:

alpha_channel_file
The name of the grayscale IFF-ILBM file to use as the alpha channel.

The *comp_opts* arguments (old format) must follow the *loader_opts* above.

Be sure you do not delete the `.ALPHA` and `ALPHA` files from the **Loaders2** directory. The ALPHA loader needs both of these files to operate. These files are placed in this directory by the installation program.

If load was	RC	ADPRO_RESULT
successful (not using `NOPAD`)	0	undefined
successful (using `NOPAD`)	5	undefined
unsuccessful (either case)	10	undefined

12.3.3 ANIM

The ANIM loader lets you load a single frame from IFF-ANIM animation files. It can be accessed through the main control screen and through ARexx.

ARexx Interface

Syntax: `LFORMAT "ANIM"`
`LOAD` *anim_filename* *loader_opts*

The *loader_opts* for the ANIM loader are:

(`FRAME` *n* | `COUNT` | `QUIT`)
Specify whether to load a specific frame (`FRAME`), count how many frames are available, or close the ANIM file so that other program may use it.

GUI Equivalent: **Frame Number:** for `FRAME` *n*
Count Frames for `COUNT`
`Quit` button.
Constraints: $0 \leq n$
Required: Yes

If load was	RC	ADPRO_RESULT
successful (not using NOPAD)	0	undefined
successful (using NOPAD)	5	undefined
unsuccessful (either case)	10	undefined

12.3.4 IFF

The IFF loader lets you load IFF-ILBM images. It can be accessed through the main control screen and through ARexx.

ARexx Interface

Syntax: `LFORMAT "IFF"`
`LOAD` *image_to_load loader_opts*

The IFF loader supports the `NOPAD` keyword for *loader_opts*. If the IFF-ILBM file being read contains rendered image data (i.e., neither 24 bit-plane color nor 8 bit-plane gray), the `NOPAD` argument tells MorphPlus *not* to convert the rendered image data back into raw image data. Because this option disables the conversion on the newly loaded image into 24 bit-plane color or 8 bit-plane grayscale data, the image is loaded more quickly. This might be used from ARexx to implement a simple "slide-show" type program. This feature also makes conversions between same depth file formats much faster.

If load was	RC	ADPRO_RESULT
successful (not using NOPAD)	0	undefined
successful (using NOPAD)	5	undefined
unsuccessful (either case)	10	undefined

Note that when `ADPRO_RESULT` = 5, it means that something was loaded, but not converted into raw image data. Therefore you shouldn't check `RC` this way:

```
/* WRONG WAY */
LFORMAT "IFF"
LOAD "SOMETHING" NOPAD
IF RC ~= 0 THEN DO
   OKAY1 "FAILED!"
```

```
   EXIT
END
```

but rather:

```
/* RIGHT WAY */
LFORMAT "IFF"
LOAD "SOMETHING" NOPAD
IF RC = 10 THEN DO
   OKAY1 "FAILED!"
   EXIT
END
```

12.3.5 RIPPLE

The RIPPLE loader lets you create grayscale representations of ripples to use as alpha channel files. These alpha channels are used when compositing an image over the current image in MorphPlus. It can be accessed through the main control screen and through ARexx.

ARexx Interface

Syntax: `LFORMAT "RIPPLE"`
`LOAD "XXX"` *loader_opts*

The *loader_opts* for the RIPPLE loader are similar to the ARexx arguments for the Ripple operator, but include a few changes and additions:

CENTER *xpos ypos*

Set the location of the ripple center.

GUI Equivalent:	**Center X (Px)** and **Center Y (Px)** values.
Constraints:	$-30000 \leq xpos \leq 30000$ $-30000 \leq ypos \leq 30000$
Required:	Yes

PROPSPEED *s*

Set the propagation speed.

GUI Equivalent:	**Speed (Px/Frame)** value.
Constraints:	$1 \leq s \leq 5000$
Required:	Yes

PERIOD l

Set the length (in pixels) of one wave period.

GUI Equivalent:	**Period (Px)** value.
Constraints:	$3 \leq l \leq 5000$
Required:	Yes

AMPLITUDE h

Set the height of the wave at the wavefront.

GUI Equivalent:	**Amplitude (0-100)** value.
Constraints:	$1.0 \leq h \leq 100.0$
Required:	Yes

AMPLITUDECENTER $centerh$

Set the center gray value of the sinusoidal wave—the range of grayscale values.

GUI Equivalent:	**Amp. Cen. (0-100)** value.
Constraints:	$0.0 \leq centerh \leq 100.0$
Required:	Yes

PHASE p

Describe the point on a single wave period that specifies the starting amplitude at the ripple center.

GUI Equivalent:	**Phase (Deg)** value.
Constraints:	$0.0 \leq p \leq 360.0$
Required:	Yes

RAMP r

Describe the number of pixels (amplitude pixels) that will be added (for **Increase**) or subtracted (for **Decrease**) from the **Amplitude** value per period.

GUI Equivalent:	**Ramp (Px/Period)** value.
Constraints:	$0.0 \leq r \leq 1000.0$
Required:	Yes

LEVELOFF lh

Describe the ending amplitude of an increasing or decreasing ripple.

GUI Equivalent:	**Level Off (Amp)** value.
Constraints:	$0.0 \leq lh \leq 1000.0$
Required:	Yes

WAVETYPE *t*

Set the type of wave (amplitude behavior) to generate. The t value can be: 0 for **Constant**, 1 for **Increase**, and 2 for **Decrease.**

GUI Equivalent:	**Wave Type** rocker switch setting.
Constraints:	t = -1, **Decrease**; t = 0, **Constant** t = 1, **Increase**
Required:	Yes

FRAME *f*

Set the current instance (moment in "time") to see.

GUI Equivalent:	**Frame** value.
Constraints:	$-100000 \leq f \leq 100000$
Required:	Yes

IMAGEX *width*

Set the width (in pixels) of the resulting image.

GUI Equivalent:	**Image X (Px)** value.
Constraints:	$64 \leq width \leq 4096$
Required:	Yes

IMAGEY *height*

Set the height (in pixels) of the resulting image.

GUI Equivalent:	**Image Y (Px)** value.
Constraints:	$64 \leq height \leq 4096$
Required:	Yes

UPPERCLIP *uc*

Set the maximum value for actual wave amplitude.

GUI Equivalent:	**Upper Clip (0-100)** value.
Constraints:	$0 \leq uc \leq 100$
Required:	Yes

LOWERCLIP *lc*

Set the minimum value for actual wave amplitude.

GUI Equivalent:	**Lower Clip (0-100)** value.
Constraints:	$0 \leq lc \leq 100$
Required:	Yes

NEWWAVE

Specifies the end of a ripple definitions. All of the above parameters, up to the previous **NEWWAVE** argument or **OPERATOR "RIPPLE"** arguments (for the first ripple definition, will be used to create a new ripple.

GUI Equivalent:	The Ripple Crosshair.
Constraints:	None
Required:	Yes. End each ripple definition with this argument.

NOTE: The first wave you specify must list **all** *arguments and end with the* `NEWWAVE` *keyword. Subsequent wave definitions only need to specify those values that have changed since the previous definition; all other attributes will be inherited.*

See the Ripple operator reference (Section 12.5.14) for an example of specifying multiple ripples.

The *comp_opts* arguments (new format) must follow the *loader_opts* above.

If load was	RC	ADPRO_RESULT
successful (not using NOPAD)	0	undefined
successful (using NOPAD)	5	undefined
unsuccessful (either case)	10	undefined

12.3.6 TEMP

The TEMP loader lets you replace the raw image data in the primary image buffer with the raw image data currently in the temporary buffer. This loader can be executed from the main cotrol screen or through its ARexx interface.

■ ARexx Interface

Syntax:
```
LFORMAT "TEMP"
LOAD "XXX"
```

There are no *loader_opts* for the TEMP loader.

The *comp_opts* arguments (new format) must follow the "XXX" parameter.

If load was	RC	ADPRO_RESULT
successful (not using NOPAD)	0	undefined
successful (using NOPAD)	5	undefined
unsuccessful (either case)	10	undefined

12.3.7 UNIVERSAL

The UNIVERSAL loader will try to load in a particular image file, using its filename and embedded information to detect and invoke the correct loader. It can be accessed through the main control screen and through ARexx.

ARexx Interface

Syntax: **LFORMAT "UNIVERSAL"**
LOAD *image_to_load*

The UNIVERSAL loader requires some special discussion. You can use the UNIVERSAL loader from ARexx. However, since you don't know which loader will ultimately be used to load the image, you cannot make use of any loader options.

If load was	RC	ADPRO_RESULT
successful (not using NOPAD)	0	undefined
successful (using NOPAD)	5	undefined
unsuccessful (either case)	10	undefined

12.4 Savers

Savers can be selected from the main screen or through their ARexx interfaces. For specific information, read each saver's control panel and ARexx interface sections.

12.4.1 General Information

ARexx Interface

Save Format

Syntax: **SFORMAT** (1)
SFORMAT *format* (2)

The first form of the command fills the **ADPRO_RESULT** field with the name of the currently selected Save Format, if results are requested.

The second form of the command sets the Save Format to the specified string. MorphPlus will qualify the supplied string as being either "ANIM", "HAME", or "IFF".

If module	RC	ADPRO_RESULT
exists	0	the *previously* selected Save Format
does not exist	10	"Unknown Module"

Refer to Section 2.5.1 and Chapter 6 for more information concerning Save Formats.

Calling a Saver

Syntax: `SAVE` *filename type [saver_opts]*
This command invokes a Saver with the specified arguments.

The Saver which will be invoked is specified with the `SFORMAT` command.

A filename and save type are required arguments. The type must be one of the following strings: `RAW`, `IMAGE`, or `SCREEN`. Specifying `RAW` will attempt to save the raw image data, if available. Specifying `IMAGE` will attempt to save the rendered image data, if available.

Specifying `SCREEN` will attempt to save only as much rendered image data as will fit in the currently defined screen type (and, will save only a screen sized portion of the image if it is larger than the current screen type). Screen size is defined by the currently set screen width and height. Also, the upper left hand corner of the portion of the image to be saved is determined by scrolling the image by hand. From ARexx, you cannot automatically offset the portion of the image to be saved.

Each particular saver may have its own restrictions as to which type of data may be saved. These restrictions are defined in each saver's ARexx Interface reference section.

Remember that if you only have rendered image data, you can turn it back into raw image data by saving the rendered image data to disk and then reloading it.

The *saver_opts* are saver specific options. The currently defined saver-specific arguments are defined in each saver's ARexx Interface reference section.

12.4.2 ANIM

The ANIM saver can be controlled through a control panel or ARexx.

ARexx Interface

Syntax: **SFORMAT "ANIM"**
SAVE *anim_filename type saver_opts*

The ANIM saver may be instructed to save either **IMAGE** or **SCREEN** data *type*.

The ANIM saver's *saver_opts* include:

(IMAGE | SCREEN)
Set the type of rendered data to save.

GUI Equivalent:	**Save Type** cycle gadget value.
Constraints:	None
Required:	Yes

APPEND | WRAPUP | TRUNCATE n
Describe the operation to perform. **APPEND** adds the current rendered data to the end of the ANIM. **WRAPUP** appends the first two frames to the end of the ANIM *and* closes the file. **TRUNCATE** (takes an integer argument) either reduces the file to n frames (for positive n) or chops $-n$ frames (for negative n).

GUI Equivalent:	**Append**, **Wrap Up**, and **Truncate** buttons on the control panel. The n value is equivalent to the value specified in the Truncate control panel's input field.
Constraints:	$0 \leq n$
Required:	Yes

QUIT
Close the file so that other programs can access it.

GUI Equivalent:	**Quit** button on the control panel.
Constraints:	None
Required:	No

ARexx Interface

Syntax: **SFORMAT "HAME"**
SAVE *filename* **"IMAGE" "TODISK"** (1)
SAVE *filename* **"IMAGE" "TODISK"** *screen_type* (2)
SAVE "XXX" "IMAGE" "DISPLAY" (3)
SAVE "XXX" "IMAGE" "DISPLAY" *ticks* (4)
SAVE "XXX" "IMAGE" "DISPLAY" *ticks*
screen_type (5)

The HAME saver can only save **IMAGE** data.

The first form saves the currently rendered HAME data to the file specified by *filename*. If *filename* is NULL, then the file requester will pop up to allow

the user to manually select the name of the file in which to save the data. The screen settings used will be what ever had been last used.

The second form saves the currently rendered HAME data to the file specified by *filename*. If *filename* is NULL, then the file requester will pop up to allow the user to manually select the name of the file in which to save the data. The number specified by *screen_type* is used to determine what type of screen to construct. This number can be assembled using the values contained in Table 12.1. Note that the VGA and Super Hi-Res modes are not allowed.

The third form allows you to display the image data, rather than save it to disk. If the "DISPLAY" keyword is present, then the supplied filename is ignored. The image is displayed until the user clicks the left mouse button. The screen settings used to construct the display will be the same ones last used.

The fourth form will display the image data, but for a specific amount of time. In this form, you can specify the number of *ticks* (fiftieths of a second) to hold the image on-screen. The value specified by *ticks* may be either 0 or between 1 and 65535. If the value is 0, then the displayed image will remain on-screen until the left mouse button is clicked.

The fifth form will display the image data on a specific type of screen for a specific amount of time. The *ticks* argument is the same value as described in the fourth form. The *screen_type* argument is the same as described in the second form.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

For more information about preparing rendered image data for use with the HAME, please see Section 6.2.

12.4.3 IFF

The IFF saver can be controlled through a control panel or its ARexx interface.

ARexx Interface

Syntax: `SFORMAT "IFF"`
`SAVE` *image_filename type*

The IFF saver may be instructed to save `RAW`, `IMAGE`, or `SCREEN` data *type*.

The IFF saver does not have any *saver_opts*.

12.4.4 TEMP

The TEMP saver lets you copy the raw image data in the primary image buffer to the temporary image buffer. It can be controlled through the main MorphPlus control screen or through its ARexx interface.

ARexx Interface

Syntax: `SFORMAT "TEMP"`
`SAVE "XXX"`

The TEMP saver save command places the raw image data into the temporary image buffer.

The TEMP saver does not have any *saver_opts*.

12.5 Operators

All of the operators can be controlled through an operator-specific control panel (or screen of panels) or through their ARexx interfaces.

12.5.1 General Information

ARexx Interface

Syntax: `OPERATOR` *operator_name* [*operator_args*]

The `OPERATOR` command is a general purpose front-end to run-time loadable MorphPlus operators. Operators are independent programs typically performing an image processing function which modifies the raw and/or rendered image data currently available.

As given above, the generalized syntax of the `OPERATOR` command requires the operator name to be given as the command's first argument. Additional arguments may be required depending upon the particular operator being called.

Note that there is no ARexx command for the **Execute Op** button found below the Operator selector (on the main screen). Instead, each call to the `OPERATOR` command both selects the specified operator and causes it to execute immediately.

The following sections detail the operators provided as standard with MorphPlus.

12.5.2 Apply_Map

The Apply_Map operator does not bring up any control panel of its own. Once you select this operator, a meter window will appear and the operation will start immediately. It can also be started through its ARexx interface.

ARexx Interface

Syntax: `OPERATOR "APPLY_MAP"`

This operator causes any raw image data to be replaced by the raw image data after filtering through the current color controls.

If raw data is	RC	ADPRO_RESULT
available	0	undefined
not available	10	undefined

Please see Section 7.1 for more information.

12.5.3 Blur

The Blur operator can be controlled through its control panel (that opens on MorphPlus' screen) or through its ARexx interface.

ARexx Interface

Syntax: `OPERATOR "BLUR"` *cw th* (1)
`OPERATOR "BLUR"` *cw th* `"TEST"` (2)

The first form of the operator will blur any raw data present using the supplied center weight (*cw*) and threshold (*th*).

The second form of the operator will test the supplied values by performing the appropriate calculations but not actually modify the image.

Both forms return the number of pixels which would be (in the second form) or were (as in the first form) affected by the operation in `ADPRO_RESULT`.

The *cw* value may range from 0 to 16; *th* may range from 0 to 255.

If	RC	ADPRO_RESULT
operation was successful	0	undefined
raw data is not available or values are out of range	10	undefined

Please see Section 7.2 for more information.

12.5.4 Color_To_Gray

The Color_To_Gray operator uses a control panel to specify how to do the conversion. This operator also has an ARexx interface.

Control Panel

Keyboard Equivalents	
Perform the Operation	`Shift RETURN`

ARexx Interface

Syntax: `OPERATOR "COLOR_TO_GRAY"` (1)
`OPERATOR "COLOR_TO_GRAY"` *rw gw bw* (2)

This operator converts raw color image data into raw grayscale image data.

This operator requires raw color data.

The first form of the command converts color information into grayscale using the weighting factors used by the NTSC standard.

The second form allows you to specify the weighting to be used for the red, green, and blue components of the image. Weights are scaled to a range of 0 for a zero weight to 10000 for a 100 percent weight. Weights may be negative or exceed 100 percent.

If raw color data is	RC	ADPRO_RESULT
available	0	undefined
not available	10	"Error"

For more information about the Color_To_Gray operator, please see Section 7.3.

12.5.5 Crop_Visual

The Crop_Visual operator can be controlled through a control panel on its own control screen, as well as through its ARexx interface.

Control Panel

Keyboard Equivalents	
Zooming In	z
Zip/Unzip the Control Panel	u
Perform the Operation	`Shift RETURN`

■ ARexx Interface

Syntax: `OPERATOR "CROP_VISUAL"` *width height* (1)
`OPERATOR "CROP_VISUAL"` *width height left top* (2)

This operator can be used to select and preserve a rectangular region within an image and discard all data outside the selected region.

The first form lets you specify the width and height of the cropped area using an offset of 0,0 (the upper left corner of the original image).

The second form lets you specify the width, height, and left and top offsets.

Note that if you specify offsets (second form), they cannot be negative nor can the offsets plus the width or height exceed the width or height of the original image.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.5.6 DCTV

The DCTV operator does not bring up any control panel of its own. Once you select this operator, a meter window will appear and the operation will start immediately. It can also be started through its ARexx interface.

■ ARexx Interface

Syntax: `OPERATOR "DCTV"`

This operator converts the currently loaded raw image data into rendered data usable by the DCTV display device. You can use the IFF saver to save the converted data to disk or you can use the `ADPRO_DISPLAY` command to view the DCTV data from within ARexx.

The DCTV operator takes its display size and type (i.e., interlace on or not, PAL or NTSC, etc.) from MorphPlus' current screen control settings.

If MorphPlus is currently set to render 8 colors, then a three bit-plane DCTV image will be created. If MorphPlus is set to render any other type of image (not 8 colors), then a four bit-plane DCTV image will be created.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

For more information about the DCTV operator, please see Section 7.5.

12.5.7 Define_Pxl_Aspect

The Define_Pxl_Aspect operator can be controlled through its control panel or its ARexx interface.

Control Panel

Keyboard Equivalents	
Toggle Between Input Fields	`Alt RETURN`
Perform the Operation	`Shift RETURN`

ARexx Interface

Syntax: `OPERATOR "DEFINE_PXL_ASPECT"` *arguments*

where *arguments* can be zero or more of the following (listed in any order):

`XASPECT` *xasp*
Set the horizontal pixel aspect.

GUI Equivalent: **New (X) Aspect** value.
Constraints: $1 \leq xasp \leq 240$
Required: No

`YASPECT` *yasp*
Set the vertical pixel aspect.

GUI Equivalent: **New (Y) Aspect** value.
Constraints: $1 \leq yasp \leq 240$
Required: No

`XRES` *xres*
Set the horizontal resolution (DPI).

GUI Equivalent: **X DPI** value.
Constraints: $1 \leq xres \leq 9999$
Required: No

YRES *yres*

Set the vertical resolution (DPI).

GUI Equivalent:	**Y DPI** value.
Constraints:	$1 \leq yres \leq 9999$
Required:	No

This operator allows you to define the current image's pixel aspect and resolution information. If no arguments are given, then the current aspect and resolution information will be returned in a string (ADPRO_RESULT) of the following form:

xasp yasp xres yres width height

where *xasp* and *yasp* are the x and y aspect of the pixels in the currently defined image. The *xres* and *yres* values are the currently defined x and y resolutions. The *width* and *height* values are the width and height of the image in pixels. You can determine the size (in inches) that this image is asking to be by dividing the width or height values by the *xres* or *yres* fields, respectively.

If	RC	ADPRO_RESULT
no values are given	0	the *current* pixel aspect and resolution information
values are correct	0	the *previous* pixel aspect and resolution information
any of the values are out of range or do not make sense	non-zero	undefined

12.5.8 Gray_To_Color

The Gray_To_Color operator does not bring up any control panel of its own. Once you select this operator, a meter window will appear and the operation will start immediately. It can also be started through its ARexx interface.

■ ARexx Interface

Syntax: `OPERATOR "GRAY_TO_COLOR"`

This operator converts grayscale raw image data into raw color image data. The resulting raw image has identical red, green and blue components. Please see Section 7.7 for more information.

If	RC	ADPRO_RESULT
successful	0	undefined
raw grayscale image data is not present when this operator is executed or there is not enough memory available to perform the conversion	10	undefined

12.5.9 Horizontal_Flip

The Horizontal_Flip operator does not bring up any control panel of its own. Once you select this operator, a meter window will appear and the operation will start immediately. It can also be started through its ARexx interface.

ARexx Interface

Syntax: `OPERATOR "HORIZONTAL_FLIP"`

This operator flips the available raw data horizontally about its center.

This operator requires raw data.

If	RC	ADPRO_RESULT
successful	0	undefined
no raw data is available	10	"Error"

For more information about the Horizontal_Flip operator, please see Section 7.8.

12.5.10 KillTemp

The KillTemp operator does not bring up any control panel of its own. Once you select this operator, a meter window will appear and the operation will start immediately. It can also be started through its ARexx interface.

ARexx Interface

Syntax: `OPERATOR "KILLTEMP"`

This operator causes any image data currently in the temporary buffer to be erased.

If raw data is	RC	ADPRO_RESULT
available	0	undefined
not available	10	undefined

Please see Section 7.9 for more information.

12.5.11 Perspective

The Perspective operator can be controlled through its control panels (shown on a separate control screen), as well as through its ARexx interface. Its control screen also has a menu bar.

Menu Bar

Project Menu	
About...	`Rt-Amiga ?`
Accept	`Rt-Amiga Y`
Quit	`Rt-Amiga Q`

Output Menu	
Open...	`Rt-Amiga O`
Close	
Reset	
Correct Input	
Correct Output	
Soft Edge	`Rt-Amiga F`

Preview Menu		
Open...		`Rt-Amiga V`
Close		
Reset		
Preview Aspect		
	No Correction	
	1:1 Correction	
	Output Correction	
Display Type		
	Grid	
	ASDG	
	Rectangle	
Fast Render		
Preview		`Rt-Amiga P`
Abort		`Rt-Amiga A`

Rotation Menu	
Open...	`Rt-Amiga R`
Close	
Reset	
Rotate Point	
Center	`Rt-Amiga 5`
Top Left	`Rt-Amiga 7`
Top Center	`Rt-Amiga 8`
Top Right	`Rt-Amiga 9`
Right Center	`Rt-Amiga 6`
Bottom Right	`Rt-Amiga 3`
Bottom Center	`Rt-Amiga 2`
Bottom Left	`Rt-Amiga 1`
Left Center	`Rt-Amiga 4`
LookAt Point	`Rt-Amiga 0`

Camera Menu	
Open...	`Rt-Amiga C`
Close	
Reset	

Windows Menu	
Open All...	`Rt-Amiga W`
Close All	`Rt-Amiga E`

Settings Menu	
Reset to Default	`Rt-Amiga D`
Last Saved	`Rt-Amiga S`

ARexx Interface

Syntax: `OPERATOR "PERSPECTIVE"` *arguments*

where *arguments* can be one or more of the following (listed in any order):

`CAMERA_POSITION` *x y z*

Set the camera position.

GUI Equivalent: **Camera Position** values.
Constraints: None
Required: Yes

FOCAL_LENGTH *length*

Set the focal length.

GUI Equivalent:	**f** value.
Constraints:	$1.0 \leq length \leq 1000.0$
Required:	No

LOOKAT_POINT *x y z*

Set the look-at point.

GUI Equivalent:	**LookAt Point** values.
Constraints:	None
Required:	No

THETA *angle*

Set the theta (θ) angle of rotation.

GUI Equivalent:	**T** value.
Constraints:	None
Required:	No

GAMMA *angle*

Set the gamma (γ) angle of rotation.

GUI Equivalent:	**G** value.
Constraints:	None
Required:	No

PHI *angle*

Set the phi (ϕ) angle of rotation.

GUI Equivalent:	**P** value.
Constraints:	None
Required:	No

SOFT_EDGE

Determine whether the image will have soft (anti-aliased) edges.

GUI Equivalent:	**Edge Type = Soft Edge**.
Constraints:	None
Required:	No

OUTPUT_SIZE *width height*

Set the dimensions of the resulting image.

GUI Equivalent:	**Width** and **Height** values.
Constraints:	$2 \leq width$, $2 \leq height$
Required:	No

OUTPUT_OFFSET *xoff yoff*

Set the offset values of the resulting image.

GUI Equivalent:	**XOffset** and **YOffset** values.
Constraints:	None
Required:	No

PIXEL_ASPECT *xasp yasp*

Set the pixel aspect of the image.

GUI Equivalent:	Aspect values internal to the image; definable in the Define_Pxl_Aspect operator.
Constraints:	$1 \leq xasp$, $1 \leq yasp$
Required:	No

INPUT_ASPECT

Determine if the input aspect should be taken into account.

GUI Equivalent:	**Input Aspect=Corr Input.**
Constraints:	None
Required:	No

OUTPUT_ASPECT

Determine if the output aspect should be taken into account.

GUI Equivalent:	**Output Aspect=Corr Output.**
Constraints:	None
Required:	No

VISIBLE_SIDE

Query if the current side of the image that the viewer is seeing is the front side.

GUI Equivalent:	None
Constraints:	None
Required:	No

(ROTATE_CENTER | ROTATE_TOP_LEFT | ROTATE_TOP_CENTER | ROTATE_TOP_RIGHT | ROTATE_RIGHT_CENTER | ROTATE_BOTTOM_RIGHT | ROTATE_BOTTOM_CENTER | ROTATE_BOTTOM_LEFT | ROTATE_LEFT_CENTER | ROTATE_LOOKAT_POINT)

Set the point of rotation.

GUI Equivalent:	**Rotate Pt** value.
Constraints:	None
Required:	No

IGNOREASPECT

Specify whether aspect correction should be done to the Preview window image.

GUI Equivalent:	**Preview Aspect = No Corr**
Constraints:	None
Required:	No

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.5.12 Refract

The Refract operator can be controlled through its control panel (shown on MorphPlus' screen), as well as through its ARexx interface.

ARexx Interface

Syntax: `OPERATOR "REFRACT"` *arguments*

where *arguments* can be one or more of the following (specified in any order):

`POOLDEPTH` *d*
Set the depth of the body of water (pool).

GUI Equivalent:	**Pool Depth** value.
Constraints:	$1.0 \leq d \leq 500.0$
Required:	Yes

`REFRACTINDEX` *i*
Set the liquid's index of refraction.

GUI Equivalent:	**Refraction Index** value.
Constraints:	$1.0 \leq s \leq 10.0$
Required:	Yes

`FILENAME` *fname*
Set the file name of the IFF-ILBM image.

GUI Equivalent:	File requester selection.
Constraints:	stringlength(*fname*) < 512
Required:	Yes

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.5.13 Rendered_To_Raw

The Rendered_To_Raw operator does not bring up any control panel of its own. Once you select this operator, a meter window will appear and the operation will start immediately. It can also be started through its ARexx interface.

ARexx Interface

Syntax: `OPERATOR "RENDERED_TO_RAW"`

This operator converts the rendered image data currently loaded in MorphPlus' memory back into raw data. The type of the raw data produced (color or

grayscale) depends upon which type of rendered image was converted. For more information about this operator, please refer to Section 7.12.

If	RC	ADPRO_RESULT
successful	0	undefined
not successful (not enough memory for the raw data)	10	undefined

For more information about the Rendered_To_Raw operator, please see Section 7.12.

12.5.14 Ripple

The Ripple operator can be controlled through its control panel (shown on a separate control screen), as well as through its ARexx interface.

Menu Bar

Project Menu	
About...	`Rt-Amiga ?`
Accept	`Rt-Amiga Y`
Quit	`Rt-Amiga Q`

ARexx Interface

Syntax: `OPERATOR "RIPPLE"` *arguments*

where *arguments* can be one or more of the following (specified in any order):

`CENTER` *xpos ypos*

Set the location of the ripple center.

GUI Equivalent: **Center X (Px)** and **Center Y (Px)** values.
Constraints: $-30000 \leq xpos \leq 30000$
$-30000 \leq ypos \leq 30000$
Required: Yes

`PROPSPEED` *s*

Set the propagation speed.

GUI Equivalent: **Speed (Px/Frame)** value.
Constraints: $1 \leq s \leq 5000$
Required: Yes

PERIOD l

Set the length (in pixels) of one wave period.

GUI Equivalent:	**Period (Px)** value.
Constraints:	$3 \leq l \leq 5000$
Required:	Yes

AMPLITUDE h

Set the height of the wave at the wavefront.

GUI Equivalent:	**Amplitude (Px)** value.
Constraints:	$1.0 \leq h \leq 500.0$
Required:	Yes

PHASE p

Describe the point on a single wave period that specifies the starting amplitude at the ripple center.

GUI Equivalent:	**Phase (Deg)** value.
Constraints:	$0.0 \leq p \leq 360.0$
Required:	Yes

RAMP r

Describe the number of pixels (amplitude pixels) that will be added (for **Increase**) or subtracted (for **Decrease**) from the **Amplitude** value per period.

GUI Equivalent:	**Ramp (Px/Period)** value.
Constraints:	$0.0 \leq r \leq 1000.0$
Required:	Yes

LEVELOFF lh

Describe the ending amplitude of an increasing or decreasing ripple.

GUI Equivalent:	**Level Off (Amp)** value.
Constraints:	$0.0 \leq lh \leq 1000.0$
Required:	Yes

WAVETYPE t

Set the type of wave (amplitude behavior) to generate. The t value can be: 0 for **Constant**, 1 for **Increase**, and 2 for **Decrease**.

GUI Equivalent:	**Wave Type** rocker switch setting.
Constraints:	$t = -1$, **Decrease**; $t = 0$, **Constant** $t = 1$, **Increase**
Required:	Yes

FRAME f

Set the current instance (moment in "time") to see.

GUI Equivalent:	**Frame** value.
Constraints:	$-100000 \leq f \leq 100000$
Required:	Yes

NEWWAVE

Specifies the end of a ripple definitions. All of the above parameters, up to the previous NEWWAVE argument or OPERATOR "RIPPLE" arguments (for the first ripple definition, will be used to create a new ripple.

GUI Equivalent:	The Ripple Crosshair.
Constraints:	None
Required:	Yes. End each ripple definition with this argument.

NOTE: The first wave you specify must list **all** *arguments and end with the* NEWWAVE *keyword. Subsequent wave definitions only need to specify those values that have changed since the previous definition; all other attributes will be inherited.*

For example, the following sequence creates four ripples, with one ripple located at opposite corners of a 320 by 200 image.

```
OPERATOR "RIPPLE",
"CENTER" 0 0 "PROPSPEED" 10 "AMPLITUDE" 15,
   "PERIOD" 10 "PHASE" 0 "RAMP" 2 "LEVELOFF" 0,
   "WAVETYPE" "-1" "FRAME" 10 "NEWWAVE",
"CENTER" 320 200 "NEWWAVE"
```

Remember that if you want to break up the arguments of a command (in this case for the OPERATOR) across several lines in your ARexx script, you must use a , (comma) to terminate each line. Notice as well that negative numbers (-1 for "WAVETYPE") must be surrounded by quotes.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.5.15 Rotate

The Rotate operator can be controlled through its control panel (shown on a separate control screen), as well as through its ARexx interface.

Menu Bar

Project Menu	
About...	`Rt-Amiga ?`
Accept	`Rt-Amiga Y`
Quit	`Rt-Amiga Q`

ARexx Interface

Syntax: `OPERATOR "ROTATE"` *arguments*

where *arguments* can be one or more of the following (specified in any order):

`CENTER` *xpos ypos*

Set the location of the rotate circle's center.

GUI Equivalent:	**Center X (Px)** and **Center Y (Px)** values.
Constraints:	$-30000 \le xpos \le 30000$, $-30000 \le ypos \le 30000$
Required:	Yes

`RADIUS` *r*

Set the radius of the rotate circle.

GUI Equivalent:	**Radius (Px)** value.
Constraints:	$5 \le r \le 15000$
Required:	Yes

`AMOUNT` *t*

Set the amount of rotation (in degrees).

GUI Equivalent:	**Amount (Deg)** value.
Constraints:	$-360 \le t \le 360$
Required:	Yes

`BLUR_RADIUS` *b*

Set the percent of the radius (starting at the outer part of the circle) that will be blurred.

GUI Equivalent:	**BRadius(0-99)** value.
Constraints:	$0 \le b \le 99$
Required:	Yes

`(QUALITY_FAST | QUALITY_HIGH)`

Set the precision of the rotate operation.

GUI Equivalent:	**Quality** rocker switch setting.
Constraints:	None
Required:	Yes

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.5.16 Scale

The Scale operator can be controlled through its control panel (shown atop MorphPlus' main screen), as well as through its ARexx interface.

ARexx Interface

Syntax: `ABS_SCALE` *abs_width abs_height* (1)
`PCT_SCALE` *pct_width pct_height* (2)

The first command scales the raw bit-map presently loaded into MorphPlus to the supplied dimensions. It is equivalent to the **Variable Enlargement / Reduction** values in the control panel.

The second command scales the raw bit-map presently loaded into MorphPlus by the supplied percentages. It is equivalent to the **Percent Enlargement / Reduction** values in the control panel.

You can specify any amount of enlargement or reduction so long as the resulting image still fits in memory.

Especially important when doing expansions, MorphPlus confirms it has enough memory available to perform the desired operation before actually starting to scale. If enough memory is not available, an error will be returned immediately.

While the MorphPlus user interface will not allow an enlargement and a reduction to be specified at the same time, the ARexx interface does. If the values you specify would cause one axis to enlarge while the other would reduce, both operations will be executed before returning to your ARexx program.

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

For more information concerning scaling, please refer to Section 7.15.

12.5.17 Sphere

The Sphere operator can be controlled through its control panels (shown on a separate control screen), as well as through its ARexx interface. Its control screen also has a menu bar.

Menu Bar

Project Menu	
About...	`Rt-Amiga ?`
Accept	`Rt-Amiga Y`
Quit	`Rt-Amiga Q`

Output Menu	
Open...	`Rt-Amiga O`
Close	
Reset	
Correct Aspect	
Soft Edge	`Rt-Amiga F`

Preview Menu		
Open...		`Rt-Amiga V`
Close		
Reset		
Preview Aspect		
	No Correction	
	1:1 Correction	
	Output Correction	
Fast Render		
Preview		`Rt-Amiga P`
Abort		`Rt-Amiga A`

Control Menu	
Open...	`Rt-Amiga C`
Close	
Reset	

Windows Menu	
Open All...	`Rt-Amiga W`
Close All	`Rt-Amiga E`

Settings Menu	
Reset to Default	`Rt-Amiga D`
Last Saved	`Rt-Amiga S`

■ ARexx Interface

Syntax: `OPERATOR "SPHERE"` *arguments*

where *arguments* can be one or more of the following (specified in any order):

"DISTORTION" *d*

Set the amount of distortion (sphering).

GUI Equivalent:	**D** value.
Constraints:	$0.0 \leq d \leq 100.0$
Required:	No

"EFFECT" *e*

Set how much effect the other settings will have on the image.

GUI Equivalent:	**%** value.
Constraints:	$0.0 \leq e \leq 100.0$
Required:	No

"CIRCLE_EFFECT" *c*

Set the circleness of the outside border.

GUI Equivalent:	**C** value.
Constraints:	$0.0 \leq c \leq 100.0$
Required:	No

"SQUARE_EFFECT" *s*

Set the squareness of the image.

GUI Equivalent:	**S** value.
Constraints:	$0.0 \leq s \leq 100.0$
Required:	No

"OUTPUT_SIZE" *width height*

Set the dimensions of the resulting image.

GUI Equivalent:	**Width** and **Height** values.
Constraints:	$2 \leq width$, $2 \leq height$
Required:	No

"PIXEL_ASPECT" *xasp yasp*

Set the pixel aspect of the image.

GUI Equivalent:	Aspect values internal to the image; definable in the Define_Pxl_Aspect operator.
Constraints:	$1 \leq xasp$ $1 \leq yasp$
Required:	No

"CORRECT_ASPECT"

Determine whether to correct for the pixel aspect.

GUI Equivalent:	**Aspect Correction** cycle gadget setting.
Constraints:	None
Required:	No

`"SOFT_EDGE"`

Determine whether the image will have soft (anti-aliased) edges.

GUI Equivalent:	**Edge Type = Soft Edge.**
Constraints:	None
Required:	No

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.5.18 Twirl

The Twirl operator can be controlled through its control panel (shown on a separate control screen), as well as through its ARexx interface.

Menu Bar

Project Menu	
About...	`Rt-Amiga ?`
Accept	`Rt-Amiga Y`
Quit	`Rt-Amiga Q`

ARexx Interface

Syntax: `OPERATOR "TWIRL"` *arguments*

where *arguments* can be one or more of the following (specified in any order):

`CENTER` *xpos ypos*

Set the location of the twirl circle's center.

GUI Equivalent:	**Center X (Px)** and **Center Y (Px)** values.
Constraints:	$-30000 \leq xpos \leq 30000$, $-30000 \leq ypos \leq 30000$
Required:	Yes

`RADIUS` *r*

Set the radius of the twirl circle.

GUI Equivalent:	**Radius (Px)** value.
Constraints:	$5 \leq r \leq 15000$
Required:	Yes

AMOUNT t

Set the amount of twirl (in degrees).

GUI Equivalent:	**Amount (Deg)** value.
Constraints:	$-360 \leq t \leq 360$
Required:	Yes

BLUR_RADIUS b

Set the percent of the radius (starting at the outer part of the circle) that will be blurred.

GUI Equivalent:	**BRadius(0-99)** value.
Constraints:	$0 \leq b \leq 99$
Required:	Yes

(QUALITY_FAST | QUALITY_HIGH)

Set the precision of the twirl operation.

GUI Equivalent:	**Quality** rocker switch setting.
Constraints:	None
Required:	Yes

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.5.19 Vertical_Flip

The Vertical_Flip operator automatically performs a flip, only showing the meter window.

ARexx Interface

Syntax: `OPERATOR "VERTICAL_FLIP"`

This operator flips the available raw data vertically about its center.

This operator requires raw data.

If	RC	ADPRO_RESULT
successful	0	undefined
no raw data is available	10	"Error"

For more information about the Vertical_Flip operator, please see Section 7.18.

12.5.20 Warp

The Warp operator can be controlled through its control panels (shown on a separate control screen), as well as through its ARexx interface. Its control screen also has a menu bar.

Control Panel

Keyboard Equivalents (Control Screen)	
Vectors	`1`
Start Points Only	`2`
End Points Only	`3`
Delete Vector(s) (not placed in Clipboard)	`Del`
Multiple Selection of Vectors	`s`
Multiple Creation of New Vectors	`n`
Multiple Creation of New Points	`m`
Create Edge Between Vectors	`,`
Multiple Creation of Connected Vectors	`n ,`
Multiple Creation of Connected Points	`m ,`
Toggle Between Primary and Secondary Highlight Color	``` ` ``` (back tick)
Flash Selected Vectors	`L`
Zoom In (2:1)	`+`
(3:1)	`Shift +`
(4:1)	`Alt +`
(All the way)	`Ctrl +`
Zoom Out (2:1)	`-`
(3:1)	`Shift -`
(4:1)	`Alt -`
(All the way)	`Ctrl -`
Full Size	`\`

Keyboard Equivalents (Control Screen — continued)	
Scroll the magnified image	
(small steps)	`cursor`
(screen-size steps)	`Shift cursor`
(large steps)	`Alt cursor`
(to the far end)	`Ctrl cursor`
Abort the Movement, Creation, and Pasting of Vectors	right mouse button

NOTE: Most of the menu items that have menu shortcuts (i.e., **`Rt-Amiga`** *keys) can also be selected without holding down the* **`Rt-Amiga`** *key. The only exceptions are the* **Copy** *and* **Paste** *commands, which require the* **`Rt-Amiga`** *key.*

Keyboard Equivalents (Group Options Control Panel)	
Edit Motion Curve	`m`
Edit Transparency Curve	`t`
Toggle Interpolation Method	`l`
Toggle Time Indicator	`f`
Add Control Point	+
Remove Control Point	`Del`
Copy Selected Control Point to Clipboard	`c`
Paste From Control Point in Clipboard	`v`
Edit the Previous Group in the List	`,`
Edit the Next Group in the List	`.`
Abort the Movement and Creation of Curve Control Points	right mouse button
Use a Pre-Saved Curve Definition	`o`
Save the Current Curve Definition	`a`
Close the Control Panel	`e` or `y`

Menu Bar (Main Screen)

Project Menu	
Options...	`Rt-Amiga T`
About...	`Rt-Amiga ?`
Accept	`Rt-Amiga Y`
Quit	`Rt-Amiga Q`

Groups Menu		
Erase...		
Include...		
Save All As...		
Select Members		`Rt-Amiga B`
Visibility		
	All On	`Rt-Amiga {`
	All Off	`Rt-Amiga }`
	Choose...	`Rt-Amiga '`
Options...		`Rt-Amiga R`
Depth Control...		

Vectors Menu		
New Vector		`Rt-Amiga N`
New Point		`Rt-Amiga M`
Cut		`Rt-Amiga X`
Copy		`Rt-Amiga C`
Paste		`Rt-Amiga V`
Erase		
Open...		
Include...		
Save All As...		
Select All Visible		`Rt-Amiga [`
Select All Connected		`Rt-Amiga K`
Deselect All		`Rt-Amiga ]`
Flip Selected		`Rt-Amiga J`
Add Selected To Group...		`Rt-Amiga I`
Visibility		
	Show All	`Rt-Amiga (`
	Hide Selected	`Rt-Amiga )`
	Flash Selected	`Rt-Amiga L`
	Flash Hidden	`Rt-Amiga H`

Settings Menu		
Vectors Display		
	As Vectors	`Rt-Amiga 1`
	As Start Points	`Rt-Amiga 2`
	As End Points	`Rt-Amiga 3`
	With Vector Numbers	
Hidden Vectors		
	Invisible	
	Ghosted	
Hidden Edges		
	Visible	
	Invisible	
End Points		
	Large	
	Small	
Zoom		
	In	`Rt-Amiga +`
	Out	`Rt-Amiga -`
	Full Size	`Rt-Amiga \`
Preferences...		

■ Menu Bar (Group Options)

Project Menu	
Open...	`Rt-Amiga O`
Save As...	`Rt-Amiga A`
Close	`Rt-Amiga E`

Edit Menu	
Copy	`Rt-Amiga C`
Paste	`Rt-Amiga V`
Add Point	`Rt-Amiga M`
Remove Point	`Rt-Amiga R`

Other Menu		
Interpolation Method		
	Linear	`Rt-Amiga L`
	Spline	`Rt-Amiga S`
Display Time As		
	Frames	`Rt-Amiga F`
	Percentages	`Rt-Amiga P`

ARexx Interface

Syntax: `OPERATOR "WARP"` *arguments*

where *arguments* can be one or more of the following (specified in any order):

`PERCENT` *pct*

Set how much all ungrouped vectors will be warped.

GUI Equivalent: Default group's motion curve level at the current frame.
Constraints: None
Required: No

`FRAME` *framenum totalframes*

Specify which frame (*framenum*) of the warp (total of *totalframes*) to generate.

GUI Equivalent: **This Frame (B)** and **Total Frames** input fields.
Constraints: None
Required: No

`ANTI_ALIASING`

Specify whether anti-aliasing will be done.

GUI Equivalent: **AntiAliasing** checkbox.
Constraints: None
Required: No

DO_TRANSPARENCY
Specify whether to create a transparency.

GUI Equivalent:	**Generate Alpha Channel (M)** checkbox.
Constraints:	None
Required:	No

(QUALITY_FAST | QUALITY_HIGH)
Precision of the warp operation.

GUI Equivalent:	**Precision** cycle gadget.
Constraints:	None
Required:	No

(SLIDING | CONVEX_HULL | COOKIE_CUT | FIXED)
Specify how the outer edges or borders will behave.

GUI Equivalent:	**Outer Edge Handling** cycle gadget.
Constraints:	None
Required:	No

(HARD_EDGES | SOFT_EDGES)
Specify the edges of areas folding over other areas.

GUI Equivalent:	**Edges** cycle gadget.
Constraints:	None
Required:	No

(FORWARD | REVERSE)
Specifies the direction of all vectors. If FORWARD (the default value), all source points will be warped to their corresponding destination points. If REVERSE, the movement will go from destination to source.

GUI Equivalent:	None
Constraints:	None
Required:	No

VECTOR x_1 y_1 x_2 y_2
Define a new vector.

GUI Equivalent:	Start and end locations of a vector.
Constraints:	$0 \leq x_1$, $0 \leq y_1$, $0 \leq x_2$, $0 \leq y_2$
Required:	No

VECTORFILE *vectorfname*
Set the name of vector definition file to include.

GUI Equivalent:	**Include** menu item (**Vectors** menu).
Constraints:	stringlength(*vectorfname*) < 80
Required:	No

GROUPFILE *groupfname*

Set the name of group definition file to include.

GUI Equivalent:	**Include** menu item (**Groups** menu).
Constraints:	stringlength(*groupfname*) < 80
Required:	No

EDGE v_1 v_2

Define an edge between two vectors.

GUI Equivalent:	Edge between two vectors.
Constraints:	$0 \le v_1 \le numvectors$, $0 \le v_2 \le numvectors$, $v_1 \ne v_2$
Required:	No

If operation was	RC	ADPRO_RESULT
successful	0	undefined
not successful	10	undefined

12.6 Morph

Morph can be controlled through its control panels (shown on a separate control screen). Its control screen also has a menu bar.

Control Panels

Keyboard Equivalents (Onion Skin VUI)	
Vectors	1
Start Points Only	2
End Points Only	3
Delete Vector(s) (not placed in Clipboard)	Del
Multiple Selection of Vectors	s
Multiple Creation of New Vectors	n
Multiple Creation of New Points	m
Create Edge Between Vectors	,
Multiple Creation of Connected Vectors	n ,
Multiple Creation of Connected Points	m ,

Keyboard Equivalents (Onion Skin VUI — continued)	
Toggle Between Primary and Secondary Highlight Color	` (back tick)
Flash Selected Vectors	`L`
Show	
Only Source Image	`Ctrl c`
More of Source Image (large steps)	`Alt c`
More of Source Image (medium steps)	`Shift c`
More of Source Image (small steps)	`c`
Show	
Only Destination Image	`Ctrl v`
More of Destination Image (large)	`Alt v`
More of Destination Image (medium)	`Shift v`
More of Destination Image (small)	`v`
Show Source Image with Start Points Only	`d`
Show 50% Mix of Source and Dest. Images with Both Start and End Points	`f`
Show Destination Image with End Points Only	`g`
Zoom In (2:1)	`+`
(3:1)	`Shift +`
(4:1)	`Alt +`
(All the way)	`Ctrl +`
Zoom Out (2:1)	`-`
(3:1)	`Shift -`
(4:1)	`Alt -`
(All the way)	`Ctrl -`
Full Size	\
Scroll the magnified image	
(small steps)	`cursor`
(screen-size steps)	`Shift cursor`
(large steps)	`Alt cursor`
(to the far end)	`Ctrl cursor`
Abort the Movement, Creation, and Pasting of Vectors	right mouse button

Keyboard Equivalents (Group Options Control Panel)	
Add Control Point	`m`
Remove Control Point	`r`
Abort the Movement and Creation of Curve Control Points	right mouse button

Almost all control panels in Morph have their own keyboard equivalents. The knowledge of them outside of their panel is of little use, so they will not be listed here.

Menu Bar (Main Screen)

Project Menu		
New		
	Moving Morph...	
	Still Morph...	
	Moving Warp...	
	Still Morph...	
Open...		`Rt-Amiga O`
Close		`Rt-Amiga E`
Save		`Rt-Amiga S`
Delete...		
Refresh...		
Options...		`Rt-Amiga T`
About...		`Rt-Amiga ?`
Accept		`Rt-Amiga Y`
Quit		`Rt-Amiga Q`

Frame Menu	
Next	`Rt-Amiga >`
Previous	`Rt-Amiga <`
Specify...	`Rt-Amiga /`

Groups Menu		
Erase...		
Include...		
Save All As...		
Select Members		`Rt-Amiga B`
Visibility		
	All On	`Rt-Amiga {`
	All Off	`Rt-Amiga }`
	Choose...	`Rt-Amiga '`
Options...		`Rt-Amiga R`
Depth Control...		
Tween...		
Animate...		

Vectors Menu	
New Vector	`Rt-Amiga N`
New Point	`Rt-Amiga M`
Cut	`Rt-Amiga X`
Copy	`Rt-Amiga C`
Paste	`Rt-Amiga V`
Erase	
Open...	
Include...	
Save All As...	
Select All Visible	`Rt-Amiga [`
Select All Connected	`Rt-Amiga K`
Deselect All	`Rt-Amiga ]`
Flip Selected	`Rt-Amiga J`
Add Selected To Group...	`Rt-Amiga I`
Visibility	
Show All	`Rt-Amiga (`
Hide Selected	`Rt-Amiga )`
Hide Common	`Rt-Amiga $`
Hide Source Only	`Rt-Amiga %`
Hide Destination Only	`Rt-Amiga ^`
Flash Selected	`Rt-Amiga L`
Flash Hidden	`Rt-Amiga H`
Vector Type	
Common To Both	`Rt-Amiga 4`
Source Only	`Rt-Amiga 5`
Destination Only	`Rt-Amiga 6`

Settings Menu		
Vectors Display		
	As Vectors	`Rt-Amiga 1`
	As Start Points	`Rt-Amiga 2`
	As End Points	`Rt-Amiga 3`
	With Vector Numbers	
Hidden Vectors		
	Invisible	
	Ghosted	
Hidden Edges		
	Visible	
	Invisible	
End Points		
	Large	
	Small	
Zoom		
	In	`Rt-Amiga +`
	Out	`Rt-Amiga -`
	Full Size	`Rt-Amiga \`
Create Icons		
Preferences...		
Load Settings...		
Save Settings		
Save Settings As...		

Applications Menu
MorphPlus/ADPro
FRED
Workbench

Menu Bar (Group Options)

Project Menu	
Open...	`Rt-Amiga O`
Save As...	`Rt-Amiga A`
Close	`Rt-Amiga E`

Edit Menu	
Copy	`Rt-Amiga C`
Paste	`Rt-Amiga V`
Add Point	`Rt-Amiga M`
Remove Point	`Rt-Amiga R`

Other Menu		
Interpolation Method		
	Linear	Rt-Amiga L
	Spline	Rt-Amiga S
Display Time As		
	Frames	Rt-Amiga F
	Percentages	Rt-Amiga P

12.7 FRED

FRED can be controlled through its control panels (shown on a separate control screen). Its control screen also has a menu bar.

Control Screen

When you start FRED, an empty light gray screen will appear. This is the backdrop, or control, screen. Sequence windows open atop this screen.

The backdrop screen is active when its backdrop window (its border visible along the left, right, and bottom of the screen) is active—the border is drawn in the highlight (greenish) color. A sequence window is active when its window borders are drawn in the highlight color (and the screen's borders are not).

A few of the control panels in FRED have keyboard equivalents. The knowledge of them outside of their panel is of little use, so they will not be listed here.

Menu Bar

Project Menu	
New...	Rt-Amiga N
Open...	Rt-Amiga O
closE...	Rt-Amiga E
Save	Rt-Amiga S
Save As...	Rt-Amiga A
About...	Rt-Amiga ?
Switch To...	
Quit	Rt-Amiga Q

The Options and AnimOps menus are only available when the backdrop screen is active.

Options Menu		
Screen Depth		
	4 Bit-Planes	
	5 Bit-Planes	
	6 Bit-Planes	
	7 Bit-Planes	
	8 Bit-Planes	
Set Text Options...		`Rt-Amiga '`

AnimOps Menu
Compositor

The Edit and Miscellaneous menus are only available when a sequence window is active.

Edit Menu	
Insert Image(s)...	`Rt-Amiga I`
Insert Range...	`Rt-Amiga G`
Change Duration...	`Rt-Amiga T`
Image Info...	`Rt-Amiga F`
Select Range	`Rt-Amiga R`
Select All	`Rt-Amiga [`
DeSelect All	`Rt-Amiga ]`
Cut	`Rt-Amiga X`
Copy	`Rt-Amiga C`
Paste	`Rt-Amiga V`
Reverse Paste	`Rt-Amiga` \5

Miscellaneous Menu		
Animate		
	Play Once	`Rt-Amiga 5`
	Continuous Play	`Rt-Amiga 6`
	Ping Pong	`Rt-Amiga 7`
	Entire Sequence	
	Selected Range	
Animate Speed		
	30/25 Frames Per Second	
	15/15 Frames Per Second	
	12/12.5 Frames Per Second	
	10/10 Frames Per Second	
Invoke ADPro...		`Rt-Amiga P`
Make Stamp		`Rt-Amiga M`

12.7.1 Compositor

Menu Bar

Project Menu		
New...		Rt-Amiga N
Open...		Rt-Amiga O
Close...		Rt-Amiga E
Save		Rt-Amiga S
Save As...		Rt-Amiga A
Mode		
	Define Components	Rt-Amiga 1
	Define Results	Rt-Amiga 2
About...		Rt-Amiga ?
Accept		Rt-Amiga Y
Quit		Rt-Amiga Q

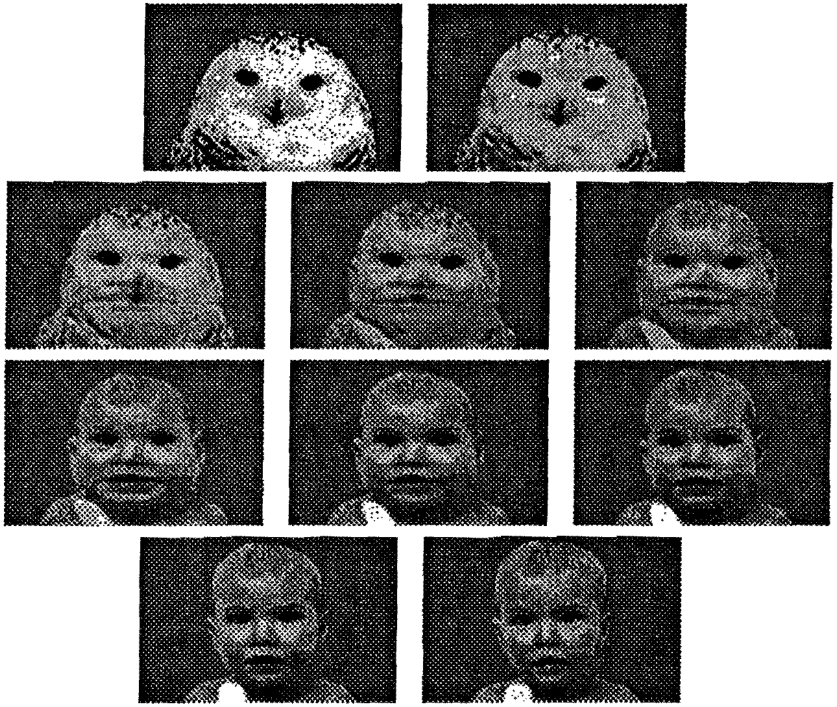
Edit Menu	
Add Sequence	
Cut	Rt-Amiga X
Copy	Rt-Amiga C
Paste	Rt-Amiga V
Erase	
Next	Rt-Amiga >
Previous	Rt-Amiga <

Appendix A

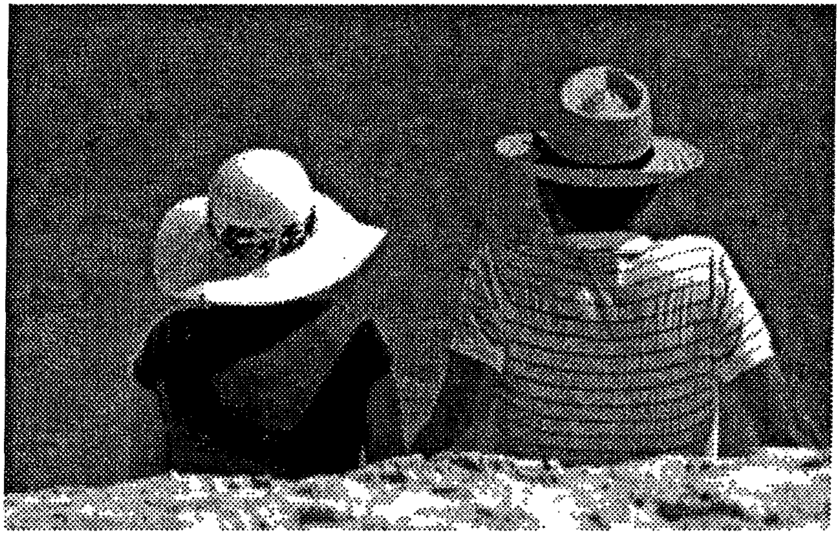
Effects Samples

This appendix contains printed samples of output from some of the operators. Included are before and after pictures of the Perspective, Refract, Ripple, Rotate, Sphere, Twirl, and Warp operators. A 10 frame morph (created with the Morph program) is also included.

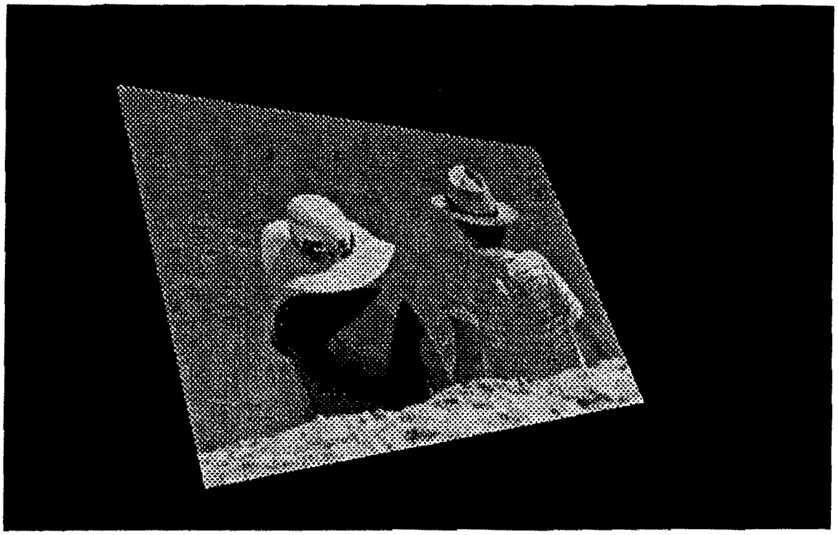
Morph AnimOp



Perspective Operator



Before



After

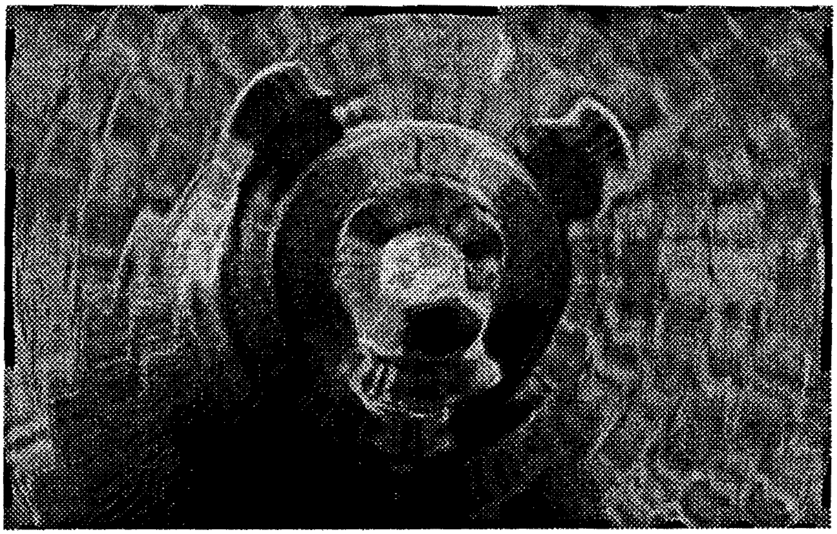
Refract Operator

Before

After

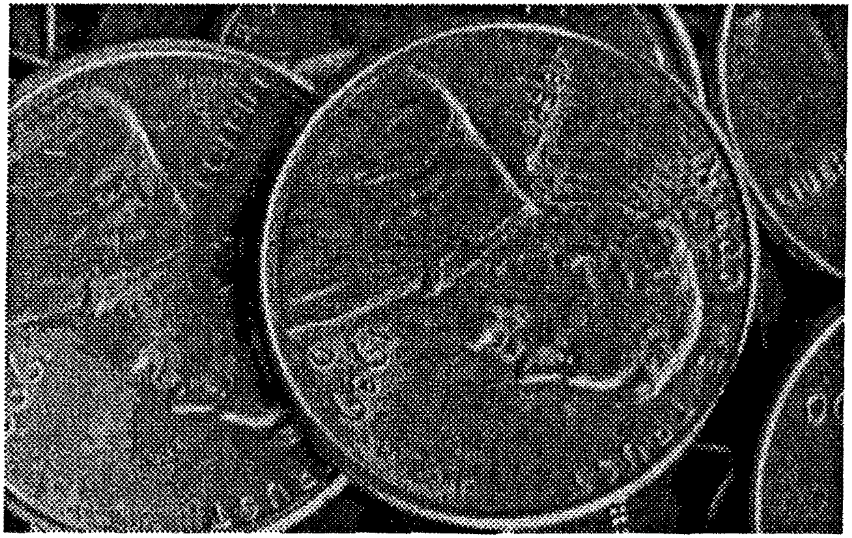
Ripple Operator

Before

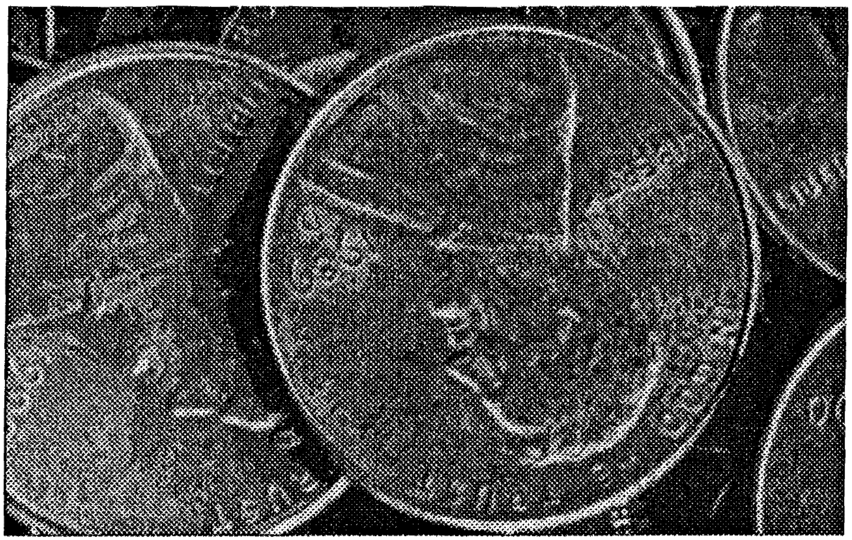


After

Rotate Operator



Before



After

Sphere Operator

Before

After

Twirl Operator

Before

After

Warp Operator

Before

After

Appendix B

Troubleshooting

This appendix describes some common problems that you might come across using MorphPlus. For each problem, a possible reason and solution are given.

Use this appendix as your initial aid whenever you come across a problem. If the problem is hardware-related (and not a problem with the MorphPlus software itself), consult the documentation that came with your hardware. If the problem relates to MorphPlus, please contact our Technical Support department (our phone number is listed in Section 1.4).

Problem: Mouse pointer imagery is getting messed up when you are holding down a key in the Morph program or Warp operator.

Cause: You are probably running a mouse blanking utility.

Solution: Disable the mouse blanker when using Morph or Warp.

Appendix C

Hints and Tips

This appendix lists some hints and tips you might find helpful when working with MorphPlus. If you find other hints and tips while working with the programs that come with this package and would like to share these with other MorphPlus users, you are more than welcome to send us a letter, call us on the phone, or leave electronic mail at any one of the various electronic information networks of which an ASDG representative is a member. If we find that your hint or tip may be helpful to other users, we will include it in future versions of the manual and post it on these information networks.

C.1 MorphPlus

The topics discussed in this section include:

- Anti-Aliasing
- Caching the Loader, Saver, and Operator Modules
- Compositing Tips
- Finer Scrolling
- Focused Color Picking
- Grayscale Balancing

- Merging a Color Image Into a Grayscale Image
- Mixing Picked and Computed Colors
- Multiple MorphPlus Configurations from the Workbench
- Reading and Writing Toaster Files
- Scaling as an Anti-Aliasing Tool
- Time Lapse and Motion Blur
- Turning Low Color Images Into Higher Res Grayscale
- Using the Warp Operator to Flatten an Extremity

Anti-Aliasing

The jaggie edges that often detract from computer generated images are called aliasing problems. The term is derived from the fact that computers divide images into little color boxes called pixels. A pixel, being of a given shape and having a well defined border with its neighboring pixels, cannot perfectly model the color information it represents. The error between what a pixel can represent and what it is supposed to represent causes the jaggies (or aliases).

It is common to want to eliminate the jaggies from your images. This process is called anti-aliasing. Note that when you anti-alias (or remove the jaggies) you must, in effect, blur the original image to some degree. In other words, to eliminate the jaggies, you must give up some amount of spatial resolution. MorphPlus has several tools to anti-alias your images and gives you control over how much actually blurring will take place.

Caching the Loader, Saver, and Operator Modules

MorphPlus can cache loaders, savers, and operators in a higher speed disk (such as `RAM:`). To make use of this feature, create an environment variable called `ADPRODIR` and set it to `RAM:`. Then create Loaders2, Savers2, and Operators2 directories. Copy into these directories the loaders, savers, or operators you will use most often.

These LSOs (Loader, Saver, and Operators) will then load from `RAM:` instead of from disk, resulting in faster performance (especially during massive batch processes).

For example, you might make a Shell script such as this (or perhaps include something like this in your user-startup):

Assume you want to cache:

- IFF and RIPPLE loaders
- IFF and ANIM saver
- Perspective operator

You would execute the following commands:

```
setenv ADPRODIR RAM:
makedir RAM:Loaders2
makedir RAM:Savers2
makedir RAM:Operators2
copy ADPRO:Loaders2/(IFF|RIPPLE) RAM:Loaders2
copy ADPRO:Savers2/(IFF|ANIM) RAM:Savers2
copy ADPRO:Operators2/Perspective RAM:Operators2
```

This mechanism also provides a way of splitting modules across multiple volumes (like floppies). To make MorphPlus run from multiple floppies (for example), create a primary floppy disk with as much of MorphPlus as can fit. Include **empty files** for all the modules which don't fit. If you don't include the empty dummy files, MorphPlus won't know about these modules.

Put the real versions of the modules which wouldn't fit on the second floppy in the appropriate directories and execute the appropriate `setenv` command.

Compositing Tips

MorphPlus' image compositing capabilities include the ability to select a specific color as being transparent. As an image is loaded during a compositing operation, each pixel is compared against the transparent color. If the pixel matches the transparent color, the new pixel is ignored and the pixel from the old image is left untouched.

This capability can be used in a two step masking process to paste an arbitrary shape from one image into another.

Suppose you had a 24 bit-plane scan of a full moon that you wish to place into a ray-tracing.

First, load the scan of the moon into MorphPlus. Render the image (with dithering) in any Amiga display mode which is compatible with your favorite paint program. We recommend 32 colors, low resolution, non-interlaced. We suggest that this rendering be made with dithering on so that you can get the best impression of what the 24 bit-plane data looks like.

Save this rendering under a different name from the original 24 bit-plane data. Do not overwrite the 24 bit-plane data as this will be needed later.

Load the rendered data into your favorite paint program. For our example, we assume Deluxe Paint IV. Bring up the image's palette. Make color register 0 completely black, that is, make its red, green, and blue components all zero. Also, make the highest color register completely white (red, green, and blue all 15's).

Select color register 0 (completely black) as your pen color and trace around the perimeter of the moon. When this is done, fill the outside of the moon completely with color register 0. In Deluxe Paint IV, this can be done easily with `Alt`-Fill. Next, fill the inside of the moon with the completely white color register. Save this image (overwriting its former self is acceptable). This is your mask.

Load the original 24 bit-plane moon image. Select the image compositing mode and then load the mask. When the image compositing control panel comes up, specify that you wish white (255, 255, 255) to be the transparent color. Specify an x and y offset of 0 so that the images completely overlap. Perform the load.

What you should now have in memory is the 24 bit-plane moon left completely untouched. However, its background should now be set completely and uniformly to black. Save this 24 bit-plane file for use later.

To complete the process load the ray-tracing. Then, in compositing mode, load the 24 bit-plane image saved in the previous step. This time, specify black (0,0,0) to be the transparent color. This instructs MorphPlus to load just the moon into the ray-tracing since the background of that image is uniformly "transparent".

Finer Scrolling

When displaying an image in a native Amiga display mode, the image can be scrolled using the four cursor keys. Each scroll step is fairly large, however. You can decrease the size of each scroll step by depressing the `Ctrl` (Control) key in conjunction with any of the four cursor keys.

Focused Color Picking

Sometimes, when rendering a 24 bit-image into a small number of colors, you might want to channel MorphPlus' color picking to give a small area of the total bit-map the best possible color choices (while sacrificing color quality elsewhere in the image).

Left alone, MorphPlus will choose the color which best matches the entire image. However, using the Crop_Visual operator, you can channel MorphPlus' color picking to optimally pick colors based upon only a small subsection of the entire image.

This can be done as follows:

- Save out the 24 bit-plane image because you will need to preserve it through the next steps.
- Using the Crop_Visual operator, crop the area of focus for color picking.
- Have MorphPlus choose the desired number of colors based upon the cropped area.
- Lock the palette.
- Reload the original image saved in the first step. Remember, the palette is locked.
- Render the entire image in the desired number of colors using the locked palette. Because you are using the locked palette based only upon the sub-region, that region will have optimal color choices within the entire image.

Grayscale Balancing

We have found that some very striking results can be had in grayscale by heavily increasing contrast and then adjusting brightness or gamma or both. While color images are best brightened using the gamma correction, we have found that adjusting the brightness of a grayscale image (using the Brightness control) will give a more natural looking result.

Merging a Color Image Into a Grayscale Image

When merging a grayscale image into a color background, you are adding an 8 bit-plane image to a 24 bit-plane background. However, to add a color image to a grayscale background, you would need to add a 24 bit-plane image to an 8 bit-plane background. Since 24 bit-planes won't fit in 8, but the ability to add a color image to a grayscale background would be nice, the Gray_To_Color operator was defined.

It takes 8 bit-plane raw image data and creates the equivalent grayscale data spread over 24 bit-planes. It does this by making each primary color equal and, thereby, gray.

To add a color image to a grayscale background you would first load the grayscale image which will become your background. Execute the Gray_To_Color operator. This causes MorphPlus to create 24 bit-plane data (which happens to be gray) from the original 8 bit-plane data.

Now, you can directly merge a color image since 24 bit-planes are now available for MorphPlus' internal use.

Mixing Picked and Computed Colors

In some instances it might be desired to pick some of the screen's colors by hand and let the computer pick the rest. For example, you might want to ensure that several specific colors are present in the rendered image.

Let's take a complicated example of a 32 color screen in which the following requirements are to be met:

- Color registers 0 through 7 are to be ignored completely.

- Color registers 8 through 15 will contain hand picked colors.
- Color registers 16 through 31 are to be picked by the computer.

This can be accomplished as follows:

First, enter the Palette control panel and set **Colors (Total)** to 32. Then set **Colors (Used)** to 16 and **Offset Color Zero** to 16. Exit the Palette control panel.

Second, from the main screen, set the **Colors** value to **CUST**, and hit the **Execute** button. The resulting image will have 16 computer picked colors in registers 16 through 31.

Next, enter the Palette control panel again. Set **Colors (Used)** to 24 and set **Offset Color Zero** to 8. Edit the palette to place the hand picked colors into registers 8 through 15 (this can be accomplished via palette loading as well).

Set the palette to **Locked** and exit the Palette control panel.

Finally, rerender the image with the **Colors** button still set to **CUST**. The resulting image will use all registers between 8 and 31 including the 8 which were hand picked and the 16 which were computer picked. It will avoid color registers 0 through 7 completely.

Using this technique you can have enormous flexibility in color composition for video titling, animation, or other video applications.

Multiple MorphPlus Configurations from the Workbench

To start MorphPlus with custom configurations, you can create "dummy" project icons that use MorphPlus as its Default Tool, but which only differ in the Tool Types specified. This will allow you to customize the MorphPlus environment.

For example, you can have one icon that launches MorphPlus with a maximum image buffer of 5,000,000 bytes (`MAXMEM=5000000`), and another one that uses up the largest contiguous chunk (no Tool Type required). Another icon can specify normal palette accuracy mode (`NOENHANCED`), and another can load a custom file of settings (`DEFAULTFILE=Work:Miscellaneous/ProgramSettings1`).

For more information on Tool Type options, please read Section 12.1.

Reading and Writing Toaster Files

NewTek's Video Toaster software can read and write 24 bit-plane IFF files. To do so, enter Toaster Paint and select the loading or saving of RGB files. The RGB format is precisely 24 bit-plane IFF.

Note that NewTek's software does not understand the color balancing chunks (CLUT chunks) placed in 24 bit-plane IFF files by many program, such as MorphPlus. See Section 7.1 for a discussion of the Apply_Map operator which transfers the information that would have been contained in the CLUT chunks to the image data itself.

Also see Section 7.7 for information about how to convert grayscale image data into color image data as the Toaster software cannot make use of 8 bit-plane grayscale IFF files.

The more efficient FrameStore format is proprietary to NewTek and has not been disclosed.

Scaling as an Anti-Aliasing Tool

MorphPlus' scaling technology employs a sophisticated bilinear interpolation engine with 64 bits of precision. In plain talk, MorphPlus' scaling is extremely good. And it can be used to anti-alias your images very simply.

If you have a large image that you need to scale down to a smaller size, simply scaling the image to the desired size will also anti-alias the image with optimal results.

On the other hand, if your image is already the appropriate size, you can scale it to, perhaps, 99 percent of its original size, and then scale it back to its original size. The resulting image will be nicely anti-aliased with minimal loss.

Time Lapse and Motion Blur

Image Number	Mix Percentage	Contribution of Each Image
1	-	-
2	50	1/2
3	33	1/3
4	25	1/4
5	20	1/5
n	$100/n$	$1/n$

Table C.1: Mix percentages to produce a time lapse effect.

MorphPlus' compositing ability with variable mixing can be used to create a time lapse photography effect when working with multiple frames of an animation. This is done by mixing multiple frames together in a way that causes each individual frame to contribute equally to the final image.

Table C.1 indicates the mix percentage to be used when adding each additional frame to the time lapse.

By varying the mix percentages so that the frame being added contributes the most to the resulting image, you can achieve a motion blur effect.

Turning Low Color Images Into Higher Res Grayscale

You can take a 16 color Amiga format image (for example) and turn it into a grayscale image with many more than 16 shades of gray.

Load the low color image into MorphPlus. The IFF Loader will detect that the image is colored and will pad it into 24 bit-planes. Reduce the width and height of the image by more than 50 percent, then convert the image to grayscale. The resulting 8 bit-plane image will possess more than 16 gray shades.

Using the Warp Operator to Flatten an Extremity

If you want to flatten an extremity, such as a fin of a car or an ear of a face, besides the vectors that will do the flattening, you should also

use points. Add a line of connected points near the end points of the "flattening" vectors, running the full length of the extremity. This will allow the extremity to disappear, but not affect the area of the image into which it is collapsing.

C.2 Warp and Morph

The topics discussed in this section include:

- Adjusting the Double-Click Interval
- Estimating Memory Usage for Warp and/or Morph
- Speeding Up Morph Operations

Adjusting the Double-Click Interval

If you are finding that the Set Group control panel is being displayed too much as a result of double-clicking on a vector, lower the **Double-Click:** value in the Input Preferences editor. Use the **Test** button to try out the new value.

Estimating Memory Usage for Warp and/or Morph

The more sophisticated or detailed the warp or morph, the more likely you it will require more vectors and edges. Both the Warp operator and Morph program require equivalent amounts of memory to process vectors and edges. Use the following rules as a guide:

Per	Fast Mode	High Quality Mode
Vector	2K	6K
Edge	Inconsequential	

Note that these values do not take into account the memory required for the images themselves.

Speeding Up Morph Operations

As you know, the Morph program opens a high resolution interlace screen with a grayscale representation of the images to morph on it. If you have this screen as the frontmost screen, it will slow down the completion of a morph. To make the morph finish a little earlier, move MorphPlus' screen to the front.

Another thing you can do to speed up Morph is to cache the Warp operator, placing it onto a higher speed disk (such as in `RAM:`). For instructions on how to cache modules, please read "Caching the Loader, Saver, and Operator Modules" in Appendix C.1.

Index

A

A-HAM, 83, 87
A-RES, 83, 87
A2410, 69, 82
About, 3
    command, 60, 200, 201
ABS_SCALE, 253
ADProDefaults, 60, 198
ADPRO_DISPLAY, 200
ADPRO_EXIT, 201
ADPRO_TO_BACK, 202
ADPRO_TO_FRONT, 200, 201, 216–221
ADPRO_UNDISPLAY, 200
ADPRO: directory, 7, 28, 222
ALPHA, 85
ANIM, 86, 93, 235
anti-alias, 290
Apply_Map, 97, 99, 290
    ARexx control, 238
    reference, 238
ARexx, 155
    addressing MorphPlus, 155
    launching programs, 157
    port name, 155
    RC, 156, 158
    result variable, 156, 158
    results, 156
ARexx commands
    ABS_SCALE, 253
    ADPRO_DISPLAY, 200
    ADPRO_EXIT, 201
    ADPRO_TO_BACK, 202
    ADPRO_TO_FRONT, 201
    ADPRO_UNDISPLAY, 200
    BLUE, 203
    BRIGHTNESS, 203
    CONTRAST, 204
    CURRENT_MEM_SIZE, 201
    DITHER, 213
    EXECUTE, 199
    GAMMA, 204
    GETDIR, 220
    GETFILE, 217
    GETFILES, 218
    GETNUMBER, 217
    GETSTRING, 216
    GREEN, 203
    IMAGE_TYPE, 215
    LAST_LOADED_IMAGE, 214
    LAST_SAVED_IMAGE, 213
    LFORMAT, 224
    LOAD, 224
    miscellaneous, 213
    OBTAIN_ADPRO, 222
    OKAY1, 220
    OKAY2, 221
    OPERATOR, 237
    ORIENTATION, 226
    PAUSE, 216
    PCONTRAST, 208

PCT_SCALE, 253
PGETWB, 209
PIXEL_ASPECT, 221
PIXEL_RESOLUTION, 221
PLOAD, 210
POFFSET, 207
PPOKE, 209
PSAVE, 210
PSORT, 208
PSTATUS, 206
PTOTAL, 206
PUSED, 207
PWIDTH, 205
RED, 202
RELEASE_ADPRO, 222
RENDER_TYPE, 199
SAVE, 234
SCREEN_TYPE, 211
SERIAL_NUMBER, 200
SFORMAT, 233
VERSION, 200
XSIZE, 214
YSIZE, 215
ARexx loader formats
ALPHA, 227
ANIM, 227
IFF, 228
RIPPLE, 229
TEMP, 232
UNIVERSAL, 233
ARexx loader options
FORCE_PALETTE, 224
NOPAD, 27, 228
ARexx saver formats
ANIM, 235
HAME, 235
IFF, 236
TEMP, 237
ARexx saver options, 234
aspect
image, 33
pixel, 33
aspect ratio, 32

B

balancing, 61, 65, 97, 288
ARexx control, 202
brightness, 62
command, 66
contrast, 63
control panel, 66, 67
gamma, 64
BEHIND, 197, 198
BH, 197
BIX, 3
BLUE, 203
Blur, 100
ARexx control, 238
reference, 238
BRIGHTNESS, 203
brightness, 62, 99, 288
button, 13

C

check box, 14
chooser, 17
Clipboard, 45, 135, 184
CLUT chunk, 97, 290
color controls, 27, 61, 66, 99, 238
color look-up table, 67, 99
color map, 19, 20, 58, 61–66
linear, 61
color mode, 79, 82
color picking, 67, 287
Color_To_Gray, 101
ARexx control, 239
reference, 239
Colors
command, 67–69, 77, 82
colors

available, 32
offset color zero, 70
total, 69
used, 70
command line interface, 7, 8, 10
Comp, 23
compositing, 23
alpha channel, 22
command, 23
mix level, 25
restrictions, 23
transparent color, 25
CompuServe, 3
CONTRAST, 204
contrast, 63, 64, 288
control panel, 13
control screen, 12
Crop_Visual, 29, 103, 287
ARexx control, 240
reference, 239
CURRENT_MEM_SIZE, 201
CUST, 67–69, 82, 199, 289
cycle gadget, 14

D

data flow, 32, 61
data type
conversion during load, 20
display, 20
raw, 19, 20, 27, 31, 32, 61, 66, 68, 78, 79, 81, 97, 99, 125
rendered, 19, 20, 27, 31, 32, 79, 97
requirements, 20
screen, 97
data types
internal, 19
DCTV, 104
ARexx control, 240
reference, 240
DEFAULTFILE, 198
defaults file
custom, 198
not loading it, 198
not saving it, 198
Define_Pxl_Aspect, 104, 111, 129
ARexx control, 241
reference, 241
DF, 198
display box, 15
displaying an image, 31, 60
DITHER, 213
dither, 68, 101, 286
ARexx control, 213
command, 81
types, 81
used in VUI, 29
DPI, 29
Dynamic HAM, 83, 87
Dynamic Hi-Res, 83, 87

E

edit palette, 72
EHB, 69, 87, 199, 206, 207
enhanced palette, 6, 67, 69, 209
ARexx control, 205
control, 197, 198
enlargement, 124
EXECUTE, 199
Execute
ARexx control, 199
command, 20, 32, 39, 47, 60, 72, 78, 81, 289
Execute Op, 27, 36
ARexx control, 237
Exit
ARexx control, 201
command, 60

F

file requester, 16
FRED
About..., 181
Clipboard, 184
reference, 268

G

gadget, 13
GAMMA, 204
gamma, 64, 99, 288
GEnie, 3
genlock, 70
GETDIR, 220
GETFILE, 217
GETFILES, 218
GETNUMBER, 217
GETSTRING, 216
Gray_To_Color, 23, 105
ARexx control, 242
reference, 242
GREEN, 203

H

HAM, 31, 69, 81, 87, 199, 206, 207
HAM8, 69, 199, 206, 207
HAME, 95, 235
horizontal
aspect ratio, 29
size, 79
Horizontal_Flip, 22, 59, 106
ARexx control, 243
reference, 243
HSV, 73, 75, 76

I

IFF, 87, 97, 236
internal pixel aspect, 126
image aspect, 32, 33
Image Compositing
control panel, 23
Image Information, 20
image masking, 26, 85, 286
IMAGE_TYPE, 215
input field, 14
installation, 6
Installer, 6

K

keyboard equivalents, 16
Kickstart
2.0, 4
KillTemp, 106
ARexx control, 243
reference, 243

L

Land, 22, 23
LAST_LOADED_IMAGE, 214
LAST_SAVED_IMAGE, 213
LFORMAT, 224
LOAD, 224
Load, 57, 58, 85
command, 22, 57
load format, 22, 57
loader
internal work, 26
loaders
ALPHA, 85
ANIM, 86
IFF, 87
RIPPLE, 88
TEMP, 90
UNIVERSAL, 91
Loaders2 directory, 28, 86, 227
loading, 22

M

MAXMEM, 197, 198
memory requirements
    contiguous, 4
    for color images, 4
    for grayscale images, 5
    limiting, 5, 197, 198
    reducing, 6
menu bar, 12
meter window, 18
MM, 197, 198
modules, 21
Morph, 163
    About..., 163
    closing Projects, 168
    common vector, 172
    creating Projects, 165
    deleting Projects, 169
    destination only vector, 172
    exiting, 163
    frame, 170
    keyboard shortcuts, 170, 173
    opening Projects, 168
    precision, 178
    Projects, 165
    reference, 263
    saving Projects, 168
    source only vector, 172
    vectors, 171
motion blur, 290

N

NE, 197, 198
NL, 198
NOENHANCED, 197, 198
NOLOADDEFAULTS, 198
NOPAD, 27
NOSAVEDEFAULTS, 198
NS, 198
NTSC, 80, 102
    aspect ratio, 124
    pixels, 124
    playback speed, 188
number of colors, 70, 206

O

OBTAIN_ADPRO, 222
Offset Color Zero, 70
OKAY1, 220
OKAY2, 221
OPERATOR, 237
operators
    Apply_Map, 97, 99, 238, 290
    Blur, 100, 238
    Color_To_Gray, 101, 239
    Crop_Visual, 29, 103, 239, 240, 287
    DCTV, 104, 240
    Define_Pxl_Aspect, 104, 111, 129, 241
    Gray_To_Color, 23, 105, 242
    Horizontal_Flip, 22, 59, 106, 243
    KillTemp, 106, 243
    Perspective, 23, 36, 38, 40, 106, 111, 121, 132, 244, 245
    Refract, 112, 248
    Rendered_To_Raw, 114, 248
    Ripple, 114, 249
    Rotate, 23, 121, 251, 252
    Scale, 29, 106, 123, 126, 253
    Sphere, 126, 129, 253, 255
    Twirl, 130, 256
    Vertical_Flip, 22, 59, 132, 257
    Warp, 132, 258, 261

Operators2 directory, 27, 28
ORIENTATION, 226
Orientation
    command, 22
    during a load, 22
overscan, 83
    ARexx control, 212
    horizontal, 79
    vertical, 80

P

PAL, 80, 212
    pixels, 124
    playback speed, 188
palette, 67
    256 color editor, 72
    accuracy, 69
    colors used, 70
    contrast, 71
    contrasting colors, 71
    edit, 72
    getting the Workbench colors, 78
    limited color editor, 76
    loading, 78
    locked, 58, 68
    offset of color zero, 70
    saving, 78
    sort direction, 71
    status, 68
    total colors, 69
    unlocked, 68
    use, 67
palette accuracy
    Shell option, 198
    Tool Type option, 197
PAUSE, 216
PCONTRAST, 208
PCT_SCALE, 253
Perspective, 23, 36, 38, 40, 106, 111, 121, 132
    anti-alias, 111
    ARexx control, 245
    camera controls, 107
    creating a preview, 110
    edge type, 111
    focal length, 107
    gamma, 108
    input correction, 111
    lookat point, 107
    output aspect, 111
    output controls, 111
    output correction, 111
    output dimensions, 111
    output offsets, 111
    phi, 108
    preview aspect, 110
    preview controls, 110
    preview quality, 110
    preview window, 110
    reference, 244
    rotate point, 108
    rotation controls, 108
    theta, 108
    wireframe representation, 110
PGETWB, 209
pixel aspect, 32, 33
pixel representation, 61
PIXEL_ASPECT, 221
PIXEL_RESOLUTION, 221
PLOAD, 210
POFFSET, 207
Port, 22
Portal, 3
PPOKE, 209
PSAVE, 210
PSORT, 208
PSTATUS, 206
PTOTAL, 206
PUSED, 207
PWIDTH, 205

R

radio button, 15
RC, 156, 158
RED, 202
ReDisplay
    command, 59, 60, 72, 200
reduction, 124
Refract, 112
    ARexx control, 248
    reference, 248
registration card, 3
RELEASE_ADPRO, 222
RENDER_TYPE, 199
Rendered_To_Raw, 114
    ARexx control, 248
    reference, 248
Replc, 23
resolution, 29, 79, 81, 82, 101, 104, 105, 205
REXX: directory, 157
RGB, 73, 75, 76
RGB files, 290
RIPPLE, 88
Ripple, 114
    additional control through ARexx, 116
    amplitude, 117
    amplitude behavior, 116, 119
    ARexx control, 249
    centering, 116
    crosshair, 115
    interference patterns, 114
    level off, 119
    loading definitions, 120
    period, 117
    phase, 117
    reference, 249
    ripple selection, 120
    saving definitions, 120
    speed, 117
    wavefront indicator, 120
rocker switch, 14
Rotate, 23, 121
    amount of rotation, 123
    ARexx control, 252
    blurring, 123
    center of rotation, 121
    centering, 123
    circle radius, 123
    precision, 123
    reference, 251
    Rotate Circle, 121
    when not to use, 121

S

SAVE, 234
Save, 31, 59
    command, 31, 59
save format, 31, 59
savers
    ANIM, 93
    HAME, 95
    IFF, 97
    TEMP, 98
Savers2 directory, 28
Scale, 29, 106, 123, 126
    ARexx control, 253
    reference, 253
screen
    opening behind others, 197, 198
Screen Controls, 27, 67, 79
SCREEN_TYPE, 211
scroller, 15
selling MorphPlus, 3
serial number, 3, 60
    ARexx control, 200
SERIAL_NUMBER, 200
SETCPU, 4
SFORMAT, 233
SHAM, 83, 87

Shell, 10
starting Morph, 8
starting MorphPlus, 7
starting FRED, 8
Shell options, 198
BEHIND, 198
DEFAULTFILE, 198
MAXMEM, 198
MM, 198
NE, 198
NOENHANCED, 198
NOLOADDEFAULTS, 198
NOSAVEDEFAULTS, 198
slider, 15
Sphere, 126, 129
anti-alias, 129
ARexx control, 255
aspect correction, 129
circleness, 127
creating a preview, 128
distortion, 127
edge type, 129
output aspect, 129
output controls, 128
output dimensions, 128
preview aspect, 128
preview quality, 128
preview window, 128
reference, 253
squareness, 127
total effect, 127
system requirements, 4

T

technical support, 3
TEMP, 90, 98, 237
text field, 15
time lapse, 290
Tool Type options, 197
BEHIND, 197
BH, 197
DEFAULTFILE, 198
DF, 198
MAXMEM, 197
MM, 197
NE, 197
NL, 198
NOENHANCED, 197
NOLOADDEFAULTS, 198
NOSAVEDEFAULTS, 198
NS, 198
Twirl, 130
amount of twirl, 131
ARexx control, 256
blurring, 131
center of twirl, 131
centering, 131
precision, 132
reference, 256
Twirl Circle, 130
twirl radius, 131

U

UNIVERSAL, 91
upgrades, 3
Usenet, 3
user interface
button, 13
check box, 14
chooser, 17
control panel, 13
control screen, 12
cycle gadget, 14
definitions, 11
descriptions, 11
display box, 15
file requester, 16
gadget, 13
input field, 14
keyboard equivalents, 16
menu bar, 12
meter window, 18

radio button, 15
rocker switch, 14
scroller, 15
slider, 15
text field, 15

V

VERSION, 200
version number
Morph, 163
MorphPlus, 200
FRED, 181
vertical
aspect ratio, 29
size, 80
Vertical_Flip, 22, 59, 132
ARexx control, 257
reference, 257
VGA, 80, 212
palette accuracy, 69
VUI, 29

W

Warp, 132
aborting a vector move, 136
ARexx control, 261
Clipboard, 45, 135
copying vectors, 135
creating edges, 139
creating vectors, 134
deleting vectors, 134
depth, 143
deselecting vectors, 135
displaying vectors, 135
edges, 138
flashing vectors, 136
flipping vectors, 134
keyboard shortcuts, 135
loading vectors, 138
magnification, 137
moving vectors, 136
multiple connected vectors, 139
multiple point creation, 135
multiple selection, 135
multiple vector creation, 135
pasting vectors, 135
points, 134
precision, 153
reference, 258
rubber sheet analogy, 133
saving vectors, 138
selecting connected vectors, 139
selecting vectors, 135
vector numbers, 136
vector's affect, 134
Workbench
2.0, 4
starting Morph, 8
starting MorphPlus, 7
WYSIWYG, 29

X

XSIZE, 214

Y

YSIZE, 215